separated into two groups of content having distinct domain names. While the data parsing information in this illustrated embodiment is illustrated using XML format, those skilled in the art will appreciate that such information can be specified in other manners. Lines 3–6 in Table 3 illustrate a first SiteURL with an ID of 1 that corresponds to a portion of the web site whose web pages are provided using the third-level domain name "insight.digimine.com." As is shown, two different VirtualServer logical site definitions each specify virtual web servers that can provide this group of content, with the virtual web servers using IP addresses 209.67.55.102 and 192.168.73.66 and both using port 0. As noted above, in some situations these IP addresses may correspond to two distinct physical machines. Alternately, a single machine can act as multiple virtual servers in various ways, such as having multiple IP addresses or by having different virtual servers that correspond to different port numbers for the machine (i.e., since each virtual server in the illustrated embodiment is based on a combination of an IP address and a TCP port number, a single machine can act as a first virtual server for secure HTTP communications on port number **0** and a second virtual server for normal HTTP communications can use port number **80**). The portion of the web site having the content corresponding to this first SiteURL is reached by a user selecting control **1909** on a web site web page (such as that illustrated in FIG. **19S**), and some of the web pages corresponding to this content are illustrated in FIGS. **19T–19AE**.

TABLE 3

```
<Sites>
    <Site Id="1" CookieIdentifiers="SITESERVER=," VisitTimeOut=" " TimeZoneName="GMT">
      <SiteUrl SiteUrlId="1" Name="https://insight.digimine.com" Url="/">
        <VirtualServer Id="1" IpAddress="209.67.55.102" TcpPort="0"/>
        <VirtualServer Id="2" IpAddress="192.168.73.66" TcpPort="0"/>
      </SiteUrl>
      <SiteUrl SiteUrlId="2" Name="http://www.digimine.com" Url="/"/>
    </Site>
</Sites>
```

The second SiteURL is defined in line 7 of Table 3 and corresponds to the rest of the web site content using the third-level domain name "www.digimine.com." In the illustrated embodiment, the last SiteUrl is a default that is used for any log entry that does not match an earlier SiteUrl definition, and thus this second SiteUrl does not require one or more associated combinations of IP address and port number in the illustrated embodiment. FIGS. **19A–19S** illustrate some of the web pages in this group of content. Those skilled in the art will appreciate that in other situations there could be a single domain name that corresponds to all of the content for the web site, or that the web site could be divided into more than two groups or could be divided into multiple groups of content without using distinct domain names.

In the illustrated embodiment, in addition to having a specified domain name, each of the two SiteURLs have a path designation for that domain name that limits the group of content corresponding to the SiteURL to the URLs that match the path designation. The path designation in the illustrated embodiment matches a prefix of the URL path, and since both SiteURLs include a prefix path designation of "/", the SiteURLs will match all URLs using that domain name (since all URL paths begin with a "/"). In other situations, different SiteURLs may be defined using a single domain name and different URLs. For example, a web site devoted to providing state law information might separate the web sites into 50 content sets corresponding to the 50 states, with the URLs for the content related to each state preceded by an initial URL such as "/Washington/" or "/Kansas/."

Table 4 illustrates various example data parsing information that defines types of interaction events with the example digiMine web site that are of interest. Those skilled in the art will appreciate that each web site owner may be interested in tracking information about different types of events. Conversely, web sites of similar types may often have interest in similar types of events. For example, merchant web sites that sell items will typically be interested in events related to such sales, such as adding items to a shopping cart or completing a purchase. For an informational web site such as the digiMine web site, it may be of interest when users view certain web pages or take actions such as submitting a contact form.

In the example XML event type data parsing information illustrated in Table 4, each event type of interest is specified using an EventDefinition event type definition. As is shown, each EventDefinition can have one or more defined EventDefinitionPatterns event type patterns that each includes a combination of a URLPattern URL path pattern that can match one or more URL paths, a QueryStringPattern query string pattern that can match one or more query strings, and an indication of a previously defined SiteURL. The values that are specified for each of these types of information are used to determine whether a log entry matches the EventDefinitionPattern by including corresponding information.

As an example, the EventDefinitionPattern specified in lines 3 and 4 of Table 4 will match log entries for the group of content corresponding to the previously defined SiteURL with an ID of 2 (i.e., the SiteURL defined in line 7 of Table 3) and any URL path that begins with the URL fragment "/company/contact_form.htm". This event type corresponds to a user requesting a Contact Form web page with which the user can supply their contact information to the web site. No value is supplied for the query string pattern portion of this event definition. In some embodiments, any of the three types of information specified for an EventDefinitionPattern can optionally not have a specified value, and if so will match any information of the corresponding type. Alternately, in other embodiments such a missing value could indicate that no information was allowed to be specified for that type of information (e.g., a log entry would not match this event type definition if it included any URL query string information), or different indications could be used to represent matching any information and matching no information.

TABLE 4

```
<Events>
    <EventDefinition Id="1" Name="Contact Form">
      <EventDefinitionPatterns SiteUrlId="2" UrlPattern="{prefix}=/company/contact_form.htm"
      QueryStringPattern=" "/> </EventDefinition>
    <EventDefinition Id="2" Name="Submit Contact Form">
      <EventDefinitionPatterns SiteUrlId="2" UrlPattern="{prefix}=/company/infoformsubmit.asp"
      QueryStringPattern=" "/> </EventDefinition>
    <EventDefinition Id="3" Name="Search">
      <EventDefinitionPatterns SiteUrlId="2" UrlPattern="{prefix}=/search.asp"
      QueryStringPattern="<keyword>=*"/> </EventDefinition>
    <EventDefinition Id="4" Name="Usc JSP">
      <EventDefinitionPatterns SiteUrlId="2" UrlPattern="{suffix}=.jsp"
      QueryStringPattern="<keyword>=+&<debug>=!"/> </EventDefinition>
    EventDefinition Id="5" Name="View General Counsel Bio">
      <EventDefinitionPatterns SiteUrlId="2" UrlPattern="{fn}=/company/BobBolan.htm"
      QueryStringPattern=" "/>
      <EventDefinitionPatterns SiteUrlId=" " UrlPattern="{prefix}=/search.asp"
      QueryStringPattern="<employeetype>=counsel"/>
    </EventDefinition>
          .
          .
          .
    <EventDefinition Id="400" Name="digiMine Login Attempt">
      <EventDefinitionPatterns SiteUrlId="1" UrlPattern="{prefix}=/I0033/login.asp"
      QueryStringPattern=" "/> </EventDefinition>
    <EventDefinition Id="401" Name="CompanyXYZ Login Attempt">
      EventDefinitionPatterns SiteUrlId="1" UrlPattern="{prefix}=/E004/login.asp"
      QueryStringPattern=" "/> </EventDefinition>
          .
          .
          .
</Events>
```

Those skilled in the art will appreciate that the various portions of the event type definitions, such as the URL path patterns and query string patterns, can be defined in various ways and to match many different sets of data. For example, in the illustrated embodiment URL path patterns include a specifier of what portion of a URL path is to be matched and of a value for that portion of the URL. The URL path portion indicators include the indicators "prefix," "suffix," and "fn," which match respectively the beginning, ending, or all of the URL. For example, for the previously illustrated digiMine web site, an event type that is intended to match any request for information from the company section of the web site could include a URL path pattern with a "prefix" indicator and a value of "/company/." Thus, any URL paths that begin with the static portion of "/company/" and include any following variable portion will match the pattern. Alternately, the URL path portion illustrated in lines 12–13 will match any URL path that ends with the suffix ".jsp", which corresponds to any Java Server Page ("JSP") web pages (although the specified query string pattern for the event type definition will limit the URLs that will match the overall event type definition). Those skilled in the art will appreciate that URL path patterns could be specified in a variety of other ways, such as using wild cards (e.g., "*") or regular expressions. In a similar manner to the URL path patterns, the query string patterns in the illustrated embodiment can also be defined to match various different sets of data. For example, the EventDefinitionPattern illustrated in lines 17 and 18 of Table 4 corresponds to a search functionality of the web site being invoked using a URL whose path begins with "/search.asp." While any number of query strings may be able to be supplied to the search.asp executable, this event pattern will match only query strings in which the query parameter name of "employeetype" is included and has a corresponding value of "counsel" (e.g., search asp?employeetype=counsel).

Rather than specifying an explicitly required value such as "counsel," the presence or absence of a query string name can also be specified. For example, with respect to the EventDefinitionPattern illustrated in lines 9 and 10 of the Table, the included query string pattern specifies that a query parameter name of "keyword" can optionally be present in the query string (with the optional presence indicated in the illustrated embodiment by using the "*" character). In addition, as previously noted, log entry information corresponding to specified query parameter names can be extracted and analyzed. For example, if this event pattern matches a log entry to indicate an occurrence of this event type, and the "keyword" query parameter name and corresponding value is included in query string information in that log entry, that value will be extracted and stored.

In addition to query parameter names whose presence is specified as being optional, the illustrated embodiment also allows query parameter names to be required for a match to occur (i.e., by using the "+" character) or to instead be disallowed for a match to occur (i.e., by using the "!" character). For example, the event pattern illustrated in lines 12 and 13 of Table 4 includes a required query parameter name of "keyword" and a disallowed query parameter name of "debug." Those skilled in the art will appreciate that in other embodiments query string patterns can be specified in other manners, such as by using prefixes or suffixes, or by using regular expression specifications.

In some situations, a query string may include multiple query string names that are identical, such as an example URL "search.asp?keyword=ABC& keyword=DEF&specifier=GHI." In the illustrated embodiment, this group of query parameter names can be matched with a query string pattern such as "<keyword>=+&<keyword>=*&<other-name>=!", which requires or allows the first two (but not the third) query parameter names in the query string and disallows a query parameter name that is not present. In other embodiments, a query string pattern would only match a query string if the query string pattern explicitly allowed

or required the presence of each query parameter name that is present in the query string. As it can be useful to separately track the values specified for each of the different query parameters even if they share a common name, such as when the order of the query parameter names is relevant in assigning different meanings to the corresponding values, the parser component can in some embodiments rename or map all (or all but one) of such query parameter names to have distinct names (e.g. to "keyword1" and "keyword2") for the purpose of storing the corresponding values. Thus, in this example, the parser component would store the corresponding value "ABC" from the example URL in a manner associated with the "keyword1" query parameter name so that it is distinct from the value "DEF" stored for the "keyword2" query parameter name.

In some situations, event type data parsing information can also specify sequences or series of related event types (also referred to as "funnels"). Such event type sequence definitions (not illustrated in Table 4) could be used in various ways, such as to store related event type information together, or to allow pre-calculation of various inter-event type information.

Another type of data parsing information that can be used to identify occurrences of interest relates to categories of related content that are available from a web site or other content set. Categories of related content can be identified and specified in many ways. One common type of category relates to information stored or presented in a hierarchical manner, as with the web pages of many web sites. In such situations, different hierarchy members can serve as one basis for identifying categories of related content, such as the hierarchy members lowest-level leaf node hierarchy members or the hierarchy members at all hierarchy levels of the hierarchy structure.

Table 5 provides an example of category type data parsing information that corresponds to the digiMine web pages illustrated in FIGS. **19A–19AE**. As previously noted, the digiMine web site is structured in a hierarchical manner with multiple sections, and the category data parsing information for the web site reflects that hierarchy. In particular, as is illustrated in FIG. **19A**, there are sections of the web site that can be accessed using controls **1903**, **1905**, **1907** and **1909**, with the corresponding groups of content related to services provided by digiMine, company-specific information, media information, and digiMine customer-specific information. In a corresponding manner, the category data parsing information for the web site has four top-level HierarchyMember category type definitions that begin at lines 3, 22, 45, and 59 of Table 5. In the illustrated embodiment, each HierarchyMember has a MemberName that is used to visually represent the HierarchyMember (such as in reports), a unique ID, and a unique PageKey name that indicates the hierarchical position of the HierarchyMember.

TABLE 5

```
<Hierarchy Id="1" MemberNameSeparator="&gt;">
    <HierarchyMember Id="1" MemberName="Services" PageKey="-1">
      <HierarchyMember Id="3" MemberName="Service Benefits" PageKey="-1-1">
        <PageKey Template SiteUrlId="2" Priority="98"
          BaseUrl="{prefix}=/services/servicebenefits.htm" QueryStringPattern=" "/>
          </HierarchyMember>
      <HierarchyMember Id="3" MemberName="Take the Quiz" PageKey="-1-2">
        <PageKey Template SiteUrlId="2" Priority="98" BaseUrl="{prefix}=/services/quiz.htm"
          QueryStringPattern=" "/> </HierarchyMember>
      <HierarchyMember Id="4" MemberName="How digiMine works" PageKey="-1-3">
        <PageKey Template SiteUrlId="2" Priority="98" BaseUrl="{prefix}=/services/howworks.htm"
          QueryStringPattern=" "/> </HierarchyMember>
      <HierarchyMember Id="5" MemberName="digiMine Data Enhancement Services" PageKey="-1-4">
        <PageKey Template SiteUrlId="2" Priority="98"
          BaseUrl="{prefix}=/services/enhancement.htm" QueryStringPattern=" "/>
          </HierarchyMember>
            .
            .
            .
      <PageKeyTemplate SiteUrlId="2" Priority="99" BaseUrl="{prefix}=/services/"
            QueryStringPattern=" "/>
    </HierarchyMember>
    <HierarchyMember Id="9" MemberName="Company" PageKey="-2">
      <HierarchyMember Id="10" MemberName="Management" PageKey="-2-1">
        <PageKeyTemplate SiteUrlId="2" Priority="98"
          BaseUrl="{prefix}=/company/management.htm" QueryStringPattern=" "/>
          </HierarchyMember>
      <HierarchyMember Id="11" MemberName="Careers" PageKey="-2-2">
        <HierarchyMember Id="12" MemberName="R &amp;amp; D" PageKey="-2-2-1">
          <PageKeyTemplate SiteUrlId="2" Priority="97"
            BaseUrl="{prefix}=/company/careers/rd.htm" QueryStringPattern=" "/>
            </HierarchyMember>
              .
              .
              .
        <HierarchyMember Id="16" MemberName="Legal" PageKey="-2-2-5">
          <PageKeyTemplate SiteUrlId="2" Priority="97"
            BaseUrl="{prefix}=/company/careers/legal.htm" QueryStringPattern=" "/>
            </HierarchyMember>
        <PageKeyTemplate SiteUrlId="2" Priority="98" BaseUrl="{prefix}=/company/careers/"
          QueryStringPattern=" "/>
      </HierarchyMember>
      <HierarchyMember Id="17" MemberName="Contact" PageKey="-2-3">
        <PageKeyTemplate SiteUrlId="2" Priority="98" BaseUrl="{prefix}=/company/contact.htm"
```

TABLE 5-continued

```
        QueryStringPattern=" "/> </HierarchyMember>
      <PageKeyTemplate SiteUrlId="2" Priority="99" BaseUrl="{prefix}=/company/"
            QueryStringPattern=" "/>
    </HierarchyMember>
    <HierarchyMember Id="18" MemberName="Media Center" PageKey="-3">
      <HierarchyMember Id="19" MemberName="News" PageKey="-3-1">
        <PageKeyTemplate SiteUrlId="2" Priority="98" BaseUrl="{prefix}=/mediacenter/news.htm"
          QueryStringPattern=" "/> </HierarchyMember>
            .
            .
            .
      <HierarchyMember Id="24" MemberName="Press Releases" PageKey="-3-4">
        <PageKeyTemplate SiteUrlId="2" Priority="98"
          BaseUrl="{prefix}=/mediacenter/pressreleases.htm" QueryStringPattern=" "/>
          </HierarchyMember>
      <PageKeyTemplate SiteUrlId="2" Priority="99" BaseUrl="{prefix}=/mediacenter/"
        QueryStringPattern=" "/>
    </HierarchyMember>
    <HierarchyMember Id="233" MemberName="Insight" PageKey="-4">
      <HierarchyMember Id="234" MemberName="digiMine" PageKey="-4-01">
        <HierarchyMember Id="235" MemberName="Reports" PageKey="-4-01-1">
          <HierarchyMember Id="236" MemberName="Executive Summary" PageKey="-4-01-1-1">
            <PageKeyTemplate SiteUrlId="1" BaseUrl="{prefix}=/I0033/reports/executive.asp"
              QueryStringPattern=" " Priority=" "/> </HierarchyMember>
          <HierarchyMember Id="237" MemberName="Site Traffic" PageKey="-4-01-1-2">
          <HierarchyMember Id="238" MemberName="Hourly Activity" PageKey="-4-01-1-2-1">
            <PageKeyTemplate
              SiteUrlId="1"BaseUrl="{prefix}=/I0033/reports/hourlyActivity.asp"
              QueryStringPattern=" " Priority="95"/> </HierarchyMember>
                .
                .
                .
          </HierarchyMember>
          <HierarchyMember Id="243" MemberName="Site Usage" PageKey="-4-01-1-3">
            .
            .
            .
          <HierarchyMember Id="247" MemberName="Category Analysis" PageKey="-4-01-1-3-4">
            PageKeyTemplate SiteUrlId="1"
              BaseUrl="{prefix}=/I0033/reports/storeanalysis.asp" QueryStringPattern=" ";
              Priority="95"> </HierarchyMember>
          <HierarchyMember Id="248" MemberName="Event Analysis" PageKey="-4-01-1-3-5">
            <PageKeyTemplate SiteUrlId="1"
              BaseUrl="{prefix}=/I0033/reports/eventAnalysis.asp" QueryStringPattern=" "
              Priority="95" /> </HierarchyMember>
          <HierarchyMember Id="249" MemberName="Funnel" PageKey="-4-01-1-3-6">
            <PageKeyTemplate SiteUrlId="1" BaseUrl="{prefix}=/I0033/reports/funnel.asp"
              QueryStringPattern=" " Priority="95"/> </HierarchyMember>
        </HierarchyMember>
            .
            .
            .
        <PageKeyTemplate SiteUrlId="0" Priority=" " BaseUrl="{prefix}=/I0033/reports/"
          QueryStringPattern=" "/>
      </HierarchyMember>
      <PageKeyTemplate SiteUrlId="1" Priority="98" BaseUrl="{prefix}=/I0033/"
        QueryStringPattern=" "/>
    </HierarchyMember>
    <HierarchyMember Id="260" MemberName="CompanyXYZ" PageKey="-4-02">
      <HierarchyMember Id="261" MemberName="Reports" PageKey="-4-02-1">
        <HierarchyMember Id="262" MemberName="Executive Summary" PageKey="-4-02-1-1">
          <PageKeyTemplate SiteUrlId="1" BaseUrl="{prefix}=/E004/reports/executive.asp"
            QueryStringPattern=" " Priority="96"/> </HierarchyMember>
              .
              .
              .
        </HierarchyMember>
      </HierarchyMember>
        .
        .
        .
    </HierarchyMember>
</Hierarchy>
```

Each category type definition can optionally include one or more PageKeyTemplate page type definitions that specify which log entries will match the category type definition and be considered to be part of the corresponding category. In the illustrated embodiment, the page type definitions include information similar to that previously discussed with respect to event patterns of event type definitions. For example, as shown in line 19 of the Table, the page type definition for the "Services" section category of web pages includes an indication of a previously defined SiteURL logical site definition, a BaseURL path pattern that can match one or more URL paths, and a QueryStringPattern query string pattern that can match one or more query strings. Values for each of these types of page type definition information can optionally have values specified as with event type definitions, and if so will be used to determine whether a log entry matches the page type definition. As is shown in line 19, the "Services" category page type definition includes a URL path pattern with a "prefix" indicator and a value of "/services/", with no value supplied for the QueryStringPattern. Thus, each of the web pages illustrated in FIGS. **19B–19K** would match this page type definition, and are therefore part of the corresponding "Services" category of the web site.

In some embodiments, such as the illustrated embodiment, category types can be structured in a hierarchical manner (e.g., to reflect content set items that are structured in a hierarchical manner). Each illustrated HierarchyMember category type definition can optionally be associated with one or more "children" HierarchyMembers that specify items at a next lower-level of the hierarchy. In the illustrated embodiment, the hierarchical relationship of the HierarchyMembers is illustrated both with indentation and with the PageKey values (e.g., a HierarchyMember with a PageKey of "–1-3-1" or "–1-3-5" is one hierarchy level below the HierarchyMember with a PageKey of "–1-3") As mentioned above, the hierarchy members directly below another hierarchy member in a hierarchical structure can be referred to as "children", and the hierarchy member directly above can be referred to as a "parent" (e.g., the HierarchyMember with a PageKey of "–1-3-1" is a child of the HierarchyMember with a PageKey of "–1-3").

For example, in addition to the page type definition in line 19, the Services category type definition also includes definitions in lines 4–18 for multiple next lower-level category type definitions. Each of these next lower-level category type definitions define children categories (or "subcategories") of the Services category, and have a format similar to that of the Services category type definition. For example, the "Service Benefits" category type definition defined in lines 4–6 of Table 5 corresponds to the web page illustrated in FIG. **19G**, and includes a page key value that illustrates the hierarchical relationship of itself to the Services category. In the illustrated embodiment, the URL path patterns and query string patterns for the category type definitions use the same pattern matching formats as those discussed previously with respect to the event type definitions, but those skilled in the art will appreciate that in other embodiments event type definitions can be specified in a different manner than category type definitions.

In the illustrated embodiment, the page type definition in line 19 of Table 5 includes a Priority value whose use reflects that, in the illustrated embodiment, a log entry is identified as belonging to only one category type definition. In such an embodiment, however, the log entry may match the page type definitions specified for multiple category type definitions (e.g., the web page illustrated in FIG. **19F** that has a URL path of "/services/enhancement.htm" will match not only the specific category type definition specified in lines 13–15 of Table 5 but also the more general parent category type definition whose page type definition is shown in line 19 of Table 5). Thus, if only one category type definition match is allowed, it is preferable that the "best" match be the one that is recorded for a log entry. In some embodiments the best match will be the most-specific category type definition (e.g., the matching category type definition at the lowest level of the hierarchical structure), while in other embodiments the best match may be the most-general matching category type definition. In the illustrated embodiment, the associated priority values are used to differentiate category type definitions at different levels of the hierarchy (e.g., the top-level category type definitions have a priority of 99 while the second-level category type definitions have a priority of 98). Using such information, the category type definitions can be organized before attempts at matching begin (putting either the highest priority values or the lowest priority values first), and the first category type definition whose page type definition matches the log entry can then be used as the single match.

While a log entry is allowed to match only a single category type definition in the illustrated embodiment, a log entry can be identified as being a member of each event type whose definition matches the log entry. Since a log entry will be checked against each available event type for a match in such an embodiment, it may not be necessary to provide Priority information with which to order the event types for checking. Conversely, in embodiments in which only one event type is allowed to match a log entry, or if the order in which the event types were to be matched was relevant for another reason, the EventDefinitionPatterns event patterns could similarly include priority information or other mechanisms for ordering the event type definitions in an appropriate manner. Similarly, if a log entry is allowed to match multiple category type definitions in other embodiments, and there is no other reason to order the category type definitions in a specific manner, such category type definitions may not include Priority value information.

When the parser component matches a log entry to a category type definition, it can increment various types of stored information about that category type, such as the number of page views, requests, visits, unique users, orders, revenue, etc. Similarly, the parser component can store similar types of information for event type occurrences that are noted. In addition, as previously illustrated in FIG. **19AD**, in some situations it is useful to provide information about the relationships between multiple defined categories. In some embodiments, such combinations or sequences of categories can be pre-defined, and the category data parsing information can include definitions for those category combinations or sequences to allow various information about those categories to be preprocessed. Alternately, in other situations a user can select any two or more defined categories, and the system calculates the specified category relationships dynamically. Similarly, while sequences or combinations of event types of interest can be predefined in the event data parsing information, in other situations a user can dynamically specify two or more sequences or combinations of events, and the information related to that combination or sequence of events can be dynamically generated. FIG. **19AC** provides an example of one report related to a sequence of event types.

In addition to the site, event, and category data parsing information, in some embodiments exclusion data parsing information can be specified to indicate types of log entries

that are not to be further processed. Table 6 includes various examples of types of exclusion data parsing information. For example, in lines 2 and 3, it is shown that IP addresses (or ranges of such addresses) can be specified such that requests from clients at those IP addresses are not included in the processing (e.g., the IP addresses for the machines used by internal users). Lines 3–11 indicate that log entries requesting files of specified types can also be excluded, such as those with file extensions of ".dll" (i.e., dynamic libraries) or ".gif" (i.e., image files using the GIF format). Lines 12–30 indicate that other types of URI patterns can be specified with which to exclude log entries that match the patterns, such as for specific files or for files with specified suffixes or prefixes. While not illustrated, similar exclusion patterns could be specified for query strings. In addition to the exclusion information, other parser component configuration information can also be specified (e.g., on a customer-specific basis) that modifies or sets internal parameters that affect the behavior of the parser component, as is illustrated in lines 31–40. Those skilled in the art will appreciate that a wide variety of parser component behaviors can be dynamically specified through the use of such configuration information. The Appendix section of this document provides additional details on types of information that can be specified for the parser component in one embodiment.

TABLE 6

```
<Config>
    <ConfigConstants Name="ExcludedClientIPRange" Value="209.67.55.54,209.67.55.62"/>
    <ConfigConstants Name="ExcludedClientIPRange" Value="209.67.55.98,209.67.55.126"/>
    <ConfigConstants Name="ExcludedURIPattern" Value="{FileExt}=.cdf"/>
    <ConfigConstants Name="ExcludedURIPattern" Value="{FileExt}=.css"/>
    <ConfigConstants Name="ExcludedURIPattern" Value="{FileExt}=.dll"/>
    <ConfigConstants Name="ExcludedURIPattern" Value="{FileExt}=.gif"/>
    <ConfigConstants Name="ExcludedURIPattern" Value="{FileExt}=.ico"/>
    <ConfigConstants Name="ExcludedURIPattern" Value="{FileExt}=.jpeg"/>
    <ConfigConstants Name="ExcludedURIpattern" Value="{FileExt}=.jpg"/>
    <ConfigConstants Name="ExcludedURIPattern" Value="{FileExt}=.js"/>
    <ConfigConstants Name="ExcludedURIPattern" Value="{fn}=getvroot.asp"/>
    <ConfigConstants Name="ExcludedURIPattern" Value="{fn}=logo.asp"/>
    <ConfigConstants Name="ExcludedURIPattern" Value="{fn}=nav.asp"/>
    <ConfigConstants Name="ExcludedURIPattern" Value="{fn}=nav_frames.asp"/>
    <ConfigConstants Name="ExcludedURIPattern" Value="{prefix}=/license/"/>
    <ConfigConstants Name="ExcludedURIPattern" Value="{prefix}=/pitcher"/>
    <ConfigConstants Name="ExcludedURIPattern" Value="{suffix}=/include/Chart.asp"/>
    <ConfigConstants Name="ExcludedURIPattern" Value="{suffix}=/include/ChartObject.asp"/>
    <ConfigConstants Name="ExcludedURIPattern" Value="{suffix}=/include/messageboard asp"/>
    <ConfigConstants Name="ExcludedURIPattern" Value="{suffix}=/include/report_check.asp"/>
    <ConfigConstants Name="ExcludedURIPattern" Value="{suffix}=/include/ReportFilter.asp"/>
    <ConfigConstants Name="ExcludedURIPattern" Value="{suffix}=/include/reportFunctions.asp"/>
    <ConfigConstants Name="ExcludedURIPattern" Value="{suffix}=/include/ReportQueries.asp"/>
    <ConfigConstants Name="ExcludedURIPattern" Value="{suffix}=/include/reportQueries.inc"/>
    <ConfigConstants Name="ExcludedURIPattern" Value="{suffix}=/include/Sql.asp"/>
    <ConfigConstants Name="ExcludedURIPattern" Value="{suffix}=/include/vbClientFunctions.asp"/>
    <ConfigConstants Name="ExcludedURIPattern" Value="{suffix}=/include/vbFunctions.asp"/>
    <ConfigConstants Name="ExcludedURIPattern" Value="{suffix}=/reports/Execchart asp"/>
    <ConfigConstants Name="ExcludedURIPattern" Value="{suffix}=/reportS/Execchartobject.asp"/>
    <ConfigConstants Name="HitsPtrsBufferSize" Value="20"/>
    <ConfigConstants Name="MaxLengthOutputField" Value="240"/>
    <ConfigConstants Name="QueryStringsKeyHashBuckets" Value="600"/>
    <ConfigConstants Name="QueryStringsKeyHashBuckets" Value="99"/>
    <ConfigConstants Name="RawHitsBufferSize" Value="100"/>
    <ConfigConstants Name="SuccessCodes" Value="200,304"/>
    <ConfigConstants Name="URIPairHashBuckets" Value="200"/>
    <ConfigConstants Name="UserAgentKeyHashBuckets" Value="99"/>
    <ConfigConstants Name="UserKeyBufferSize" Value="90"/>
    <ConfigConstants Name="UserKeyHashBuckets" Value="499"/>
</Config>
```

While the data parsing information in Tables 3–6 has been illustrated using XML format, those skilled in the art will appreciate that such data can be specified in a variety of other formats. Table 7 provides an example of specifying data parsing information for an example digiMine customer CompanyXYZ.com using SQL statements to add similar types of data parsing information to various database tables. FIGS. 27A–27B illustrate an example database scheme that could be used to hold such data parsing information. Those skilled in the art will appreciate that data specified in other formats, such as the XML data illustrated in Tables 3–6, could similarly be processed and stored in such database tables. As is shown by the event data parsing information in lines 51–137 of Table 7, CompanyXYZ is a merchant web site that allows purchase of items from the web site. As such, CompanyXYZ has interest in event types related to purchasing items, and lines 121–137 of the Table provide one example of defining a sequence of event types related to item purchase. While specific examples of database tables and their inter-relationships are illustrated in this example embodiment, those skilled in the art will appreciate that data parsing information could be stored in different database table data structure formats in other embodiments.

TABLE 7

```
--
-- parser configuration data for CompanyXYZ com
--
delete from PageHierarchy
delete from Page
delete from partitioncriteria
delete from HierarchyMember
delete from EventDefinitionPatterns
delete from EventDefinitionColumns
delete from EvenDefinition
delete from MemberTemplate
delete from Hierarchy
delete from PageKeyTemplate
delete from SiteQueryStrings
delete from ReferralQueryStrings
delete from SiteURL
delete from Server
delete from ServerBinding
delete from Site
delete from SiteURL VirtualServerXref
delete from VirtualServer
insert into Site(SiteID, CookieIdentifiers, SiteName) values (1, 'SITESERVER=,=,WEBTRENDS_ID=,',
    'CompanyXYZ')
insert into Server(ServerId, ServerName) values (1, 'Test1')
INSERT VirtualServer (ServerId, VirtualServerID, ServerBindingID, LogfilePrefix) VALUES (1, 1, 1, 'E002AA')
INSERT VirtualServer (ServerId, VirtualServerID, ServerBindingID, LogfilePrefix) VALUES (2, 2, 2, 'E002AB')
INSERT VirtualServer (ServerId, VirtualServerID, ServerBindingID, LogfilePrefix) VALUES (3, 3, 3, 'E002AC')
INSERT VirtualServer (ServerId, VirtualServerID, ServerBindingID, LogfilePrefix) VALUES (4, 4, 4, 'E002AD')
INSERT VirtualServer (ServerId, VirtualServerID, ServerBindingID, LogfilePrefix) VALUES (5, 5, 5, 'E002AE')
INSERT VirtualServer (ServerId, VirtualServerID, ServerBindingID, LogfilePrefix) VALUES (6, 6, 6, 'E002AF')
insert into ServerBinding(ServerBindingID, HostHeaderName, IPAddress, IPPort) values (1, 'Unknown', '0 0 0 0', '0')
insert into Hierarchy(HierarchyII), HierarchyName, HierarchyDepth, MemberNameSeparator)
    values(1, 'CompanyXYZ tabs', 3, '>')
insert into SiteURL(SiteURLId, Sitename, URL) values (1, 'CompanyXYZ com', '/')
insert into SiteURLVirtualServerXref(SiteURLIID), SiteURLID, VirtualServerID) values (1, 1, 1)
insert into PartitionCriteria(FactTable,PartitionCriteria,FactTableCurrentID)
    Values('Visit','Daily',1)
insert into PartitionCriteria(FactTable,PartitionCriteria,FactTableCurrentID)
    Values('Request','Daily',1)
declare @siteurlid int
set @siteurlid = (select SiteURLID from SiteURL where Sitename='CompanyXYZ com' and URL='/')
insert into EventDefinition(EventDefinitionID, EventName, AddrequestID, AddVisitID, AddPageID, AddReferrerID)
    values(1, 'Keyword Search', 1, 1, 1, 1)
insert into EventDefinitionPatterns(EventDefinitionID, SiteURLID, BaseURLPattern, BaseQuerystringPattern)
    values(1, @siteurlid, '{prefix}=/KeywordSearch.asp', '<keyword>=*')
insert into PartitionCriteria(FactTable,PartitionCriteria,FactTableCurrentID)
    Values('Keyword Search','Monthly',1)
insert into EventDefinition(EventDefinitionID, EventName, AddrequestID, AddVisitID, AddPageID, AddReferrerID)
    values(2, 'Power Search', 1, 1, 1, 1)
insert into EventDefinitionPatterns(EventDefinitionID, SiteURLID, BaseURLPattern, BaseQuerystringPattern)
    values(2, @siteurlid, '{prefix}=/PowerSearchResults asp', NULL)
insert into PartitionCriteria(FactTable,PartitionCriteria,FactTableCurrentID)
    Values('Power Search','Monthly',1)
insert into EventDefinition(EventDefinitionID, EventName, AddrequestID, AddVisitID, AddPageID, AddReferrerID)
    values(3, 'ViewProduct', 1, 1, 1, 1)
insert into EventDefinitionPatterns(EventDefinitionID, SiteURLID, BaseURLPattern, BaseQuerystringPattern)
    values(3, @siteurlid, '{prefix}=/product.asp', '<p>=+')
insert into PartitionCriteria(FactTable,PartitionCriteria,FactTableCurrentID)
    Values('View Product','Monthly',1)
insert into EventDefinition(EventDefinitionID, EventName, AddrequestID, AddVisitID, AddPageID, AddReferrerID)
    values(4, 'Add to Basket', 1, 1, 1, 1)
insert into EventDefinitionPatterns(EventDefinitionID, SiteURLID, BaseURLPattern, BaseQuerystringPattern)
    values(4, @siteurlid, '{prefix}=/checkout/basket.asp', NULL)
```

TABLE 7-continued

```
insert into PartitionCriteria(FactTable,PartitionCriteria,FactTableCurrentID)
    Values('Add to Basket','Monthly',1)
insert into EventDefinition(EventDefinitionID, EventName, AddrequestID, AddVisitID, AddPageID, AddReferrerID)
    values(5, 'Order Shipping and Billing', 1, 1, 1, 1)
insert into EventDefinitionPatterns(EventDefinitionID, SiteURLID, BaseURLPattern, BaseQuerystringPattern)
    values(5, @siteurlid, '{prefix}=/checkout/Purchase2ShippingBilling asp', NULL)
insert into PartitionCriteria(FactTable,PartitionCriteria,FactTableCurrentID)
    Values('Order Shipping and Billing','Monthly',1)
insert into EventDefinition(EventDefinitionID, EventName, AddrequestID, AddVisitID, AddPageID, AddReferrerID)
    values(6, 'Order Review', 1, 1, 1, 1)
insert into EventDefinitionPattern(EventDefinitionID, SiteURLID, BaseURLPattern, BaseQuerystringPattern)
    values(6, @siteurlid, '{prefix}=/checkout/Purchase3Review.asp', NULL)
insert into PartitionCriteria(FactTable,PartitionCriteria,FactTableCurrentID)
    Values('Order Review','Monthly',1)
insert into EventDefinition(EventDefinitionID, EventName, AddrequestID, AddVisitID, AddPageID, AddReferrerID)
    values(7, 'Order Confirmation', 1, 1, 1, 1)
insert into EventDefinitionPatterns(EventDefinitionID, SiteURLID, BaseURLPattern, BaseQuerystringPattern)
    values(7, @siteurlid, '{prefix}=/checkout/Purchase4Confirmation asp', NULL)
insert into PartitionCriteria(FactTable,PartitionCriteria,FactTableCurrentID)
    Values('Order Confirmation','Monthly',1)
insert into EventDefinition(EventDefinitionID, EventName, AddrequestID, AddVisitID, AddPageID, AddReferrerID)
    values(8, 'Order Status Check', 1, 1, 1, 1)
insert into EventDefinitionPatterns(EventDefinitionID, SiteURLID, BaseURLPattern, BaseQuerystringPattern)
    values(8, @siteurlid, '{prefix}=/checkout/YourOrders asp', NULL)
insert into PartitionCriteria(FactTable,PartitionCriteria,FactTableCurrentID)
    Values(Order Status Check','Monthly',1)
insert into EventDefinition(EventDefinitionID, EventName, AddrequestID, AddVisitID, AddPageID, AddRefferrerID)
    values(9, 'Login or Registration', 1, 1, 1, 1)
insert into EventDefinitionPatterns(EventDefinitionID, SiteURLID, BaseURLPattern, BaseQuerystringPattern)
    values(9, @siteurlid, '{prefix}=/checkout/frmLogin asp', NULL)
insert into PartitionCriteria(FactTable,PartitionCriteria,FactTableCurrentID)
    Values('Login or Registration','Monthly',1)
insert into EventDefinitionColumns(EventDefinitionID, EventDefinedColumnName, EventDefinitionColumnType,
    EventDefinitionColumnSize, MappingQueryStringColumns)
    values(1, 'keyword', 'varchar', 400, '<keyword>')
insert into EventDefinitionColumns(EventDefinitionID, EventDefinedColumnName, EventDefinitionColumnType,
    EventDefinitionColumnSize, MappingQueryStringColumns)
    values(3, 'productid', 'int', 4, '<p>')
exec meta_CreateFunnel
        @FunnelName = "funnel"
exec meta_FunnelElement_addEvent
        @FunnelName= "funnel",
        @EventName = 'View Product'
exec meta_FunnelElement_addEvent
        @FunnelName="funnel",
        @EventName = 'Add to Basket'
exec meta_FunnelElement_addEvent
        @FunnelName="funnel",
        @EventName = 'Order Shipping and Billing'
exec meta_FunnelElement_addEvent
        @FunnelName="funnel",
        @EventName = 'Order Review'
exec meta_FunnelElement_addEvent
        @FunnelName="funnel",
        @EventName = 'Order Confirmation'
insert into PageKeyTemplate(PageKeyTemplateID, BaseURL, SiteURLID, QueryStringPattern, PageType,
    PagekeyDefinition,Priority)
    values(1, NULL, @siteurlid, '<s>=+&<a>=+&<d>=+','department','-<s>-<a>-<d>#',1)
insert into PageKeyTemplate(PageKeyTemplateID, BaseURL, SiteURLID, QueryStringPattern, PageType,
    PagekeyDefinition,Priority)
    values(2, NULL, @siteurlid, '<s>=+&<a>=+','department','-<s>-<a>#',2)
insert into PageKeyTemplate(PageKeyTemplateID, BaseURL, SiteURLID, QueryStringPattern, PageType,
    PagekeyDefinition,Priority)
    values(3, NULL, @siteurlid, '<s>=+','department','-<s>#',3)
insert into PageKeyTemplate(PageKeyTemplateID, BaseURL, SiteURLID, QueryStringPattern, PageType,
    PagekeyDefinition,Priority)
    values(4, NULL, @siteurlid, NULL, 'department','-0#',4)
insert into HierarchyMember(HierarchyID, categoryDepth, Memberkey, SiteURLID, MemberName, MemberFullName,
    CategoryName) values (1, 1, '-50', @siteurlid, 'Outdoor Shop', 'Outdoor Shop', 'store')
insert into MemberTemplate(MemberKey, PageKeyPattern) values ('-50', '{prefix}=-50#')
insert into HierarchyMember(HierarchyID, categoryDepth, Memberkey, SiteURLID, MemberName, MemberFullName,
    CategoryName) values (1, 1, '-79', @siteurlid, 'Team Sports', 'Team Sports', 'store')
insert into MemberTemplate(MemberKey, PageKeyPattern) values ('-79', '{prefix>=-79#')
        .
        .
        .
insert into HierarchyMember(HierarchyID, categoryDepth, Memberkey, SiteURLID, MemberName, MemberFullName,
    CategoryName) values (1, 2, '-50-51', @siteurlid, 'Backpacking & Hiking', 'Outdoor Shop>Backpacking & Hiking',
    'activity')
```

TABLE 7-continued

```
insert into MemberTemplate(MemberKey, PageKeyPattern) values ('-50-51', '{prefix}=-50-51#')
                    .
                    .
                    .
exec HierarchyMember_initilize
update HierarchyMember set CategoryName='department'
update HierarchyMember set PageKey=MemberKey + '#'
update HierarchyMember set PageType='department'
insert into SiteQueryStrings(SiteID, QueryStringName, QueryStringColumnName) values (1, 's', 's')
insert into SiteQueryStrings(SiteID, QueryStringName, QueryStringColumnName) values (1, 'p', 'p')
insert into SiteQueryStrings(SiteID, QueryStringName, QueryStringColumnName) values (1, 'd', 'd')
insert into SiteQueryStrings(SiteID, QueryStringName, QueryStringColumnName) values (1, 'a', 'a')
insert into SiteQueryStrings(SiteID, QueryStringName, QueryStringColumnName) values (1, 'Brand', 'Brand')
insert into SiteQueryStrings(SiteID, QueryStringName, QueryStringColumnName) values (1, 'doc', 'doc')
insert into SiteQueryStrings(SiteID, QueryStringName, QueryStringColumnName) values (1, 'catid', 'catid')
insert into SiteQueryStrings(SiteID, QueryStringName, QueryStringColumnName) values (1, 'productid', 'productid')
insert into SiteQueryStrings(SiteID, QueryStringName, QueryStringColumnName) values (1, 'daysold', 'daysold')
insert into ReferralQueryStrings(Querystringname, QuerystringColumnName)
    select QueryStringName, QueryStringColumnName from SiteQueryStrings
insert into configconstants values ('ExcludedClientIP', '192.168 1 9')
insert into configconstants values ('ExcludedClientIP', '192 168 1.8')
insert into configconstants values ('ExcludedClientIP', '127.0 0 1')
insert into ConfigConstants values('MaxLengthOutputField', '240')
insert into ConfigConstants values('UserKeyHashBuckets', '899')
insert into ConfigConstants values('UserAgentKeyHashBuckets', '99')
insert into ConfigConstants values('QueryStringsKeyHashBuckets', '99')
insert into ConfigConstants values('URIPairHashBuckets', '200')
insert into ConfigConstants values('QueryStringsKeyHashBuckets', '600')
Exec meta_ComboTableAddEntry 'CategoryCombos', 'SPDataOutputCategory', 'Level 2 Category Combos', 'A',
    'SPFriendlyCategory'
Exec meta_EntityTableAddCategoryDepth 'CategoryCombos', 1, 2
insert into meta_DimProperty (dimname,PropertyName,PropertyDisplayName, input_PropetyName,
    input_SqlDataType_Def ,
input_ColumnNumber, SqlType, SqlType_Def,SqlType_Length, SqlType_Precision, SqlType_Scale,
    SqlType_AllowNulls,SqlType_DefaultValue, TransformationString, IsAddedToSchema, IsLookup, IsDerived,
    IsStatic, IsDaily, IsWeekly, IsMonthly, IsMultiValued, IsAggregated, IsHash, IsIdentifiable)
    Values('RegUser','UserKey','UserKey','UserKEy','Varchar(255) Null', 1, 'Varchar','Varchar(255) NULL',
    255,0,null,1,'0',null, 0,0,0,1,0,0,0,0,0,0,1)
go
Update Site
set TimeZonename ='GMT'
go
exec Tablecreationfrommetadata
go
exec meta_CreateAgrTables_reguser_Activity_by_Property
exec meta_CreateRepViews_reguser_Activity_by_Property
```

It is often the case that web sites and other content sets change in structure and content from time to time. For such changing web sites, data parsing information may have been defined for the original version of the web site and log entry information may have already been gathered for that web site. In fact, a single log file may contain entries that correspond to two or more different versions of the same web site. Unfortunately, it is often the case that the data parsing information that corresponds to one version of a web site must change in order to accurately reflect a new version of the web site. For example, the definitions for a previously existing event type or category type may change in the new version of a web site. Alternately, a previously existing event type or category type may no longer exist in the new version of the web site, and new event types of interest and category types may be present in the new web site version. Thus, it is important to be able to accurately identify the appropriate data parsing information to be used when parsing a log file and/or each log file entry.

FIG. **20** provides an example of a revised web page for the digiMine web page previously illustrated in FIG. **19**B. In particular, with respect to that web page, control **1918** has been removed in the revised web page and control **2005** has been added. This may reflect, for example, a change in the types of services offered by digiMine such that Data Enhancement services are no longer available but Data Generation (e.g., for testing purposes) services are now available.

In order to associate the appropriate data parsing information with log files or log file entries being processed, in some embodiments the data parsing information includes version information. Table 8 includes some of the data parsing information previously illustrated in Tables 3–6, but with the data parsing information modified to include version information. In particular, in the illustrated embodiment, many of the data parsing information entries include values for beginning and ending dates that define an effective date range for which the data parsing information is valid. For example, in lines 35–37 the category definition type corresponding to the digiMine data enhancement services web page illustrated in FIG. **19**F has been modified so that its effective end date ends at the day before the web site is modified. In addition, lines 38–41 illustrate a new category type definition that corresponds to the new data generation services web page that has been added to the modified web site (and is accessible via control **2005** illustrated in FIG. **20**). The beginning date of effectiveness for the new category type definition is the day on which the updated web page is put into use.

TABLE 8

```
<Sites>
    <Site Id="1" CookieIdentifiers="SITESERVER=,=" VisitTimeOut=" " TimeZoneName="GMT">
      <SiteUrl SiteUrlId="1" Name="https://insight.digimine.com" Url="/" BeginDate="02/15/00"
        EndDate=" ">
          <VirtualServer Id="1" IpAddress="209.67.55.102" TcpPort="0" BeginDate="05/01/00"
            EndDate="12/31/00"/>
          <VirtualServer Id="2" IpAddress="192.168.73.66" TcpPort="0" BeginDate="11/01/00"
            EndDate=" ">
                .
                .
                .
</Sites>
<Events>
    <EventDefinition Id="20" Name="View Data Enhancement Service Info" BeginDate=" "
      EndDate="01/31/01">
        <EventDefinitionPatterns SiteUrlId="2" UrlPattern="{fn}=/services/enhancement.htm"
          QueryStringPattern=" " BeginDate=" " EndDate="01/31/01"/> </EventDefinition>
    <EventDefinition Id="1001" Name="View Data Generation Service Info" BeginDate="02/01/01"
      EndDate=" ">
        <EventDefinitionPatterns SiteUrlId="2" UrlPattern="{fn}=/services/generation.htm"
          QueryStringPattern=" " BeginDate="02/01/01" EndDate=" "/>
    </EventDefinition>
                .
                .
                .
</Events>
<Hierarchy Id="1" MemberNameSeparator="&gt;">
    <HierarchyMember Id="1" MemberName="Services" PageKey="-1" BeginDate=" " EndDate=" ">
                .
                .
                .
      <HierarchyMember Id="5" MemberName="digiMine Enhancement Services" PageKey="-1-4">
        <PageKeyTemplate SiteUrlId="2" Priority="98"
          BaseUrl="{prefix}=/services/enhancement.htm" QueryStringPattern=" " BeginDate=" "
          EndDate="01/31/01"/> </Hierarchy.Member>
      <HierarchyMember Id="501" MemberName="digiMine Generation Services" PageKey="-1-9">
        <PageKeyTemplate SiteUrlId="2" Priority="98" BaseUrl="{prefix}=/services/generation.htm"
          QueryStringPattern=" " BeginDate="02/01/01" EndDate=" "/> </HierarchyMember>
                .
                .
                .
</Hierarchy>
<Config>
    <ConfigConstants Name="ExcludedClientIPRange" Value="209.67.55.54,209.67.55.62"
      BeginDate=" " EndDate=" "/>
                .
                .
                .
    <ConfigConstants Name="ExcludedURIPattern" Value="{FileExt}=.dll" BeginDate=" " EndDate=" "/>
                .
                .
                .
    <ConfigConstants Name="UserKeyHashBuckets" Value="499" BeginDate=" " EndDate="01/31/01"/>
    <ConfigConstants Name="UserKeyHashBuckets" Value="500" BeginDate="02/01/01" EndDate=" "/>
</Config>
```

Using the version information illustrated in Table 8, if a log file whose entries all have effective dates before "Jan. 31, 2001" is being processed by the parser component, then the parser component can use the category type definition in lines 35–37 but will not attempt to use the category type definition found in lines 38–40 (or if used, the category type definition would not match the entry due to the date discrepancy). Alternately, if all of the entries of the log file contain effective dates that are on or after "Feb. 1, 2001," then the use of these two category type definitions will be reversed. In other situations, a determination will be made for each log entry as to what data parsing information entries will be used to process that log entry.

Those skilled in the art will appreciate that version information can be specified in other manners, such as with more time detail (e.g., using minutes or seconds) or less time detail. Alternately, version information could be specified in other embodiments in manners other than with time information, such as by assigning unique version IDs to different groups of data parsing information. As long as information associated with a log file or log file entries can be used to identify the appropriate data parsing information version (e.g., if the appropriate version ID was added to the log file or to each log file entry, or was determinable in some other manner), then the parser component can identify the appropriate data parsing information entries to use. In other situations, data parsing information of different versions may be stored separately, such as by creating an entire new set of data parsing information for each new version of the web site that is created. If so, then the parser component need merely select the appropriate group of data parsing information to be used for a log entry file or a log entry. Even if data parsing information of different versions is stored together, as in illustrative Table 8, in some embodiments the parser component may separate the data parsing information entries into separate version groups before processing of the

log entries (e.g., for efficiency purposes). In addition, new versions of data parsing information can be used for reasons other than changes to a web site or other content set, such as a change in event types or category types of interest to a customer.

Those skilled in the art will also appreciate that results of parsing can be stored in various manners. In some embodiments the results from the parsing by the parser component may be stored in a manner independent of the data parsing information version, while in other embodiments version information will be made available for later analysis of the results of the parser component processing. For example, if a customer requests a report showing information that includes a category type definition such as that defined in lines 35–37 of Table 8, and the customer specifies a date range for the report that begins before Jan. 31, 2001 and ends after that date, it would be useful to indicate that the reason the data for the event after the date Jan. 31, 2001 drops to zero (presumably) is due to the new version of the web site rather than to a lack of customer interest in the digiMine data enhancement services. Alternately, reports that include such a category type definition could be limited by the user interface of the report requesting functionality to the effective dates of the category HierarchyMember.

FIG. **21** is a block diagram illustrating details of a warehouse server **260** suitable for executing an embodiment of the parser component **310**. The server includes a CPU **2105**, various I/O devices **2120**, storage **2110**, and memory **2130**. The I/O devices include a display **2121**, a network connection **2122**, a computer-readable media drive **2123**, and other I/O devices **2124**.

An embodiment of the parser component **310** is executing in memory, and it includes a Dimension Generator component **313** as well as various other components that are not illustrated. The storage includes various information to be used by the Dimension Generator component of the parser, including various data parsing information **340** and a log file **350** to be processed. The data parsing information includes various site definitions **2112**, event type definitions **2114**, category page type definitions **2116**, various log entry exclusion data **2117**, and optional definition version information **2119**. In the illustrated embodiment, the definition version information **2119** contains version information for the site definitions, event type definitions, and/or category page type definitions. As previously illustrated, in other embodiments, the version information may be specified and stored with the definition information to which it pertains rather than separately.

When the Dimension Generator component of the parser component executes, it obtains the various data parsing information from the storage, and uses it when processing the log file. Those skilled in the art will appreciate that in other embodiments some or all of the data parsing information and/or the log file may be stored on another computer system and accessed remotely. In particular, the Dimension Generator component includes a logical site identifier component **2151** that uses the stored site definition information to identify the defined site that corresponds to a log entry, a user identifier component **2152** that identifies a user corresponding to a log entry, and a URI identifier component **2153** that identifies the URI specified for each log entry. The Dimension Generator component also includes a category page type identifier component **2154** that uses the category page type definition information, as well as site and URI information, to determine one or more categories to which a log entry corresponds. Similarly, the Dimension Generator component includes an event type identifier component **2155** that uses the event type definitions, as well as site and URI information, to determine one or more events that correspond to a log entry. In the illustrated embodiment, the Dimension Generator component includes an optional version identifier component **2157** that can identify the version corresponding to a log file or a log entry, and can supply that information to other Dimension Generator components for use in identifying the appropriate definition information to be used. Those skilled in the art will appreciate that in other embodiments one or more of the other Dimension Generator components could instead include their own version identifier processing to be used to determine version information specific to that component. When the various Dimension Generator components identify information of relevance in a log entry, they can store the identified information in various parser-generated information files **2111** on the storage. Those skilled in the art will appreciate that these parser-generated information files could be stored remotely, or could be stored in another manner such as in a data base.

Those skilled in the art will also appreciate that the warehouse server **260** is merely illustrative and not intended to limit the scope of the present invention. Computer system **260** may be connected to other devices that are not illustrated, including through one or more networks such as the Internet or via the World Wide Web (WWW). In addition, the functionality provided by the illustrated Dimension Generator components may in some embodiments be combined in fewer components or distributed in additional components. Similarly, in some embodiments the functionality of some of the illustrated components may not be provided and/or other additional functionality may be available. For example, some embodiments may not include identification of users, or may not use version information. Alternately, in other embodiments some or all of the components may execute on another device and communicate with the warehouse server via inter-computer communication.

Those skilled in the art will also appreciate that, while various data parsing information and other information is illustrated as being stored before being used, these items or portions of them can be transferred between memory and other storage devices for purposes of memory management and data integrity. Some or all of the illustrated components, data and data structures may also be stored (e.g., as instructions or structured data) on a computer-readable medium, such as a hard disk, a memory, a network, or a portable article to be read by an appropriate drive. The components, data and data structures can also be transmitted as generated data signals (e.g., as part of a carrier wave) on a variety of computer-readable transmission mediums, including wireless-based and wired/cable-based mediums. Accordingly, the present invention may be practiced with other computer system configurations.

In the illustrated embodiment, systems interact over the Internet by sending HTTP messages and exchanging Web pages. Those skilled in the art will appreciate that the described techniques can also be used in various environments other than the Internet. As such, a "client" or "server" may comprise any combination of hardware or software that can interact, including computers, network devices, internet appliances, PDAs, wireless phones, pagers, electronic organizers, television-based systems and various other consumer products that include inter-communication capabilities. Communication protocols other than HTTP can also be used, such as WAP, TCP/IP, or FTP

As previously discussed, the content of a web site or other content set can often be separated into various categories,

and one manner of identifying such categories involves various manners in which the content is stored. FIGS. **22A** and **22B** illustrate various example embodiments in which category and hierarchy information can be associated with web site content. In particular, with respect to FIG. **22A**, one example is provided of a way in which the digiMine web site content could be stored in a hierarchical manner that reflects the previously discussed categories. FIG. **22A** provides a hierarchical illustration of how some of the web site information is stored, and illustrates a customer server **210** which includes a first storage **240** that includes the web site content to be served to users and a second storage **240** on which various other customer data is stored. The served content storage **240** includes various top-level directories that each correspond to different content sets, with a first content set A **2200** corresponding to the digiMine web site and a second content set B **2240** corresponding to a different web site hosted by the customer server computer.

The content set A digiMine web site includes an overviewA.htm file **2205** and various directories including a services directory **2210** and a company directory **2220**. In the illustrated embodiment, the overviewA.htm file corresponds to the home web page illustrated in FIG. **19A**. Similarly, the services directory will include the various information that is part of the Services section of the web site, and the company directory will similarly contain the information that is part of the Company section of the web site. In particular, the services directory includes various files **2211–2219** that correspond to the web pages illustrated in FIGS. **19B–19K**. Similarly, the contents of the company directory includes various files and subdirectories whose files correspond to the web pages illustrated in FIGS. **19L–19Q** As previously noted, such a hierarchical data storage structure provides one means of selecting category and hierarchy information for the web site content.

FIG. **22B** provides an alternate embodiment for storing web site content and determining category and hierarchy information for the content. In particular, in the embodiment illustrated in FIG. **22B**, the various content is stored in a database table **2260** that holds all of the contents of the digiMine web site. Each entry in the database table data structure represents a separate web page, as shown in column **2261**. In addition, each web page can be associated with a category ID in column **2262**. These category IDs correspond to various categories defined in a category hierarchy table **2250** defined for the digiMine web site. Those skilled in the art will appreciate that in other embodiments multiple categories could be assigned to each piece of content in table **2260**.

Each entry of the category hierarchy table represents a type of category of information for the digiMine web site, with a print-friendly identifier for the category shown in column **2251**. Each category includes a unique ID listed in column **2252** that corresponds to the IDs listed in column **2262** of table **2260**. In addition, in the illustrated embodiment, hierarchy information for the categories is provided via column **2253** of table **2250**, in which each category can optionally have the ID of another category listed as its parent category. Thus, for example, the top-level Services category does not have a parent category listed, but the Careers sub-category indicates that the Company category is its parent. Those skilled in the art will appreciate that any number of hierarchical levels can be specified in this manner. Similarly, in other embodiments the category parent column **2253** with hierarchy information could be removed from the table **2250**, thus providing category information without hierarchy information.

Those skilled in the art will appreciate that the web site content could be stored in other manners, and that category and/or hierarchy information could similarly be determined in other ways. For example, all of the web pages could be stored as individual files in a single directory, thus having no storage-based hierarchy information. Nonetheless, hierarchy information could be assigned to the web pages based on the contents of the web pages themselves, such as the interlinking of the web pages. For example, since the overviewA.htm file contains links to overview files related to services and company information, the overviewA.htm file could be selected to be higher in the hierarchy than the overview files for the service and company sections of the web site.

FIG. **23** is a flow diagram illustrating an embodiment of the Identify Page Type routine **2300**. In the illustrated embodiment, the routine identifies a log file to be parsed, retrieves various category data parsing information related to the log file including version information if available, and then processes each log entry in the log file using the appropriate data parsing information. FIG. **11** previously illustrated an alternate technique for identifying page type information for a single log entry at a time.

The routine begins at step **2305** where an indication is received of a customer whose log file is to be parsed. The routine continues to step **2310** to retrieve category type definition information for the customer including version information if available. In the illustrated embodiment each category type definition has at most one page type definition, but those skilled in the art will appreciate that in other embodiments multiple page type definitions can be associated with each category type definition. The routine then continues to step **2315** to optionally separate the retrieved definitions into version groups based on the version information if it is available. In the illustrated embodiment, this separation is performed once (e.g., as an efficiency measure) such that for any date and time of a log entry in the log file, the routine can easily identify the appropriate category type definitions that are applicable to that date and time. Those skilled in the art will appreciate that in alternate embodiments the appropriate definitions could be identified dynamically for each log entry. Alternately, in some embodiments the retrieved category type definition information may already be separated into separate version groups. If it is possible to determine from the information received in step **2305** that a subset of the version groups will apply to all of the log entries in the log file, the routine could discard (or not initially retrieve) the definitions that are not in those version groups.

After step **2315**, the routine continues to step **2320** to optionally organize the definitions in each version group if appropriate, such as based on priority if priority information is available for the different category type definitions (or their page type definitions). Alternately, other criteria could be used to order the definitions. This ordering can be important for various reasons, such as if processing for a log entry stops after the first matching category type definition is identified. The routine then continues to step **2325** to receive an indication of the next log entry from the customer's log file, beginning with the first. In some embodiments, the indication that is received in step **2305** is actually the first log entry from the log, and if so, step **2325** will be skipped during this first pass so that the first entry will be processed. The routine then continues to step **2330** to select the appropriate definition version group to process the log entry.

In step **2335**, the next definition in the version group is selected, beginning with the first. The routine continues to

step **2337** to retrieve the site definition specified by the selected version group definition. In step **2340** it is determined if the log entry matches the retrieved site definition (if any is specified), URL path pattern for the selected definition (if any is specified), and query string pattern for the selected definition (if any is specified). If so, the routine continues to step **2345** to store one or more indications of the occurrence of the selected category type in the appropriate manner, including storing any relevant information from the log entry. After step **2345**, the routine continues to step **2350** to determine if multiple category page type definitions can be matched to each log entry. In some embodiments, this could be specifiable as part of the data parsing information.

If multiple definitions are allowed in step **2350**, or if the selected definition does not match the log entry in step **2340**, the routine continues to step **2355** to determine if there are more category type definitions in the selected version group. If so, the routine returns to step **2335** to select the next definition in the version group for processing. If multiple definitions are not allowed per log entry in step **2350**, or if there are not more definitions in the selected version group in step **2355**, the routine instead continues to step **2360** to determine if there are more log entries to be processed. If so, the routine returns to step **2325** to select the next log entry for processing, and if not the routine continues to step **2365** to determine if there are more log files to process. If there are more log files, the routine continues to step **2305**, and if not then the routine continues to step **2395** and ends.

FIG. **24** is a flow diagram illustrating an embodiment of the Identify Events routine **2400**. In the illustrated embodiment, the routine receives an indication of a customer whose log file is to be parsed, retrieves event type definitions related to the log file including version information if available, and uses the retrieved information to process the log file. FIG. **12** previously illustrated an alternate technique for identifying event type information for a single log entry at a time.

The routine begins at step **2405** where an indication is received of a customer whose log file is to be parsed. The routine continues to step **2410** to retrieve event type definition information for the customer, and in step **2415** retrieves information for each event pattern defined for the event type definitions including any version information if available. Those skilled in the art will appreciate that in other embodiments the event type definition information and event pattern information would be stored together. The routine next continues to step **2420** to optionally separate the retrieved definitions into version groups based on the version information if it is available. In the illustrated embodiment, this separation is performed once (e.g. as an efficiency measure) such that for any date and time of a log entry in the log file, the routine can easily identify the appropriate event type definitions that are applicable to that date and time. Those skilled in the art will appreciate that in alternate embodiments the appropriate definitions could be identified dynamically for each log entry. Alternately, in some embodiments the retrieved event type definition information may already be separated into separate version groups. If it is possible to determine from the information received in step **2405** that a subset of the version groups will apply to all of the log entries in the log file, the routine could discard (or not initially retrieve) the definitions that are not in those version groups.

After step **2420**, the routine continues to step **2425** to optionally organize the definitions in each version group if appropriate, such as based on priority if priority information is available for the different event type definitions (or their event patterns). Alternately, other criteria could be used to order the definitions. This ordering can be important for various reasons, such as if processing for a log entry stops after the first matching event type definition is identified. The routine then continues to step **2430** to receive an indication of the next log entry from the customer's log file, beginning with the first. In some embodiments, the indication that is received in step **2405** is actually the first log entry from the log, and if so, step **2430** will be skipped during this first pass so that the first entry will be processed. The routine then continues to step **2435** to select the appropriate definition version group to process the log entry. In step **2440**, the next event type definition in the version group is selected, beginning with the first. The routine then selects in step **2445** the next event pattern for the selected event type definition, beginning with the first. The routine continues to step **2450** to retrieve the site definition specified by the selected event pattern. In step **2455** it is determined if the log entry matches the retrieved site definition (if any is specified), URL path pattern for the selected definition (if any is specified), and query string pattern for the selected definition (if any is specified). If the log entry does not match, the routine continues to step **2460** to determine if there are more event patterns for the selected event type, and if so returns to step **2445** to select the next event pattern.

If the log entry does match, however, the routine continues to step **2465** to store one or more indications of the occurrence of the selected event type in the appropriate manner, including storing any relevant information from the log entry. After step **2465**, the routine continues to step **2470** to determine if multiple event page type definitions can be matched to each log entry. In some embodiments, this could be specifiable as part of the data parsing information. In the illustrated embodiment, however, while a log entry may match multiple event types, each log entry is only allowed to match one event pattern per event type. Those skilled in the art will appreciate that in other embodiments multiple event patterns could be matched per event type.

If multiple definitions are allowed in step **2470**, or if the selected event pattern does not match the log entry in step **2460**, the routine continues to step **2475** to determine if there are more event type definitions in the selected version group. If so, the routine returns to step **2440** to select the next event type definition in the version group for processing. If multiple definitions are not allowed to match each log entry in step **2470**, or if there are not more definitions in the selected version group in step **2475**, the routine instead continues to step **2480** to determine if there are more log entries to be processed. If so, the routine returns to step **2430** to select the next log entry for processing, and if not the routine continues to step **2485** to determine if there are more log files to process. If there are more log files, the routine continues to step **2405**, and if not then the routine continues to step **2495** and ends.

FIG. **25** is a flow diagram illustrating an embodiment of the Generate Interaction Data Report routine **2500**. The routine receives an indication to generate a report or otherwise provide information about previously processed interaction data and provides the appropriate report or information Those skilled in the art will appreciate that in alternate embodiments, rather than retrieving stored information from prior processing, the interaction data could be dynamically processed after the information request is received.

The routine begins at step **2505** where an indication is received to generate a report that includes information about specified types of interaction data over a specified date range. The routine continues to step **2510** to determine if

event type data is requested to be included in the report, and if so continues to step **2515** to retrieve stored information on occurrences of those event types that occurred during the specified date range. After step **2515** or if no event type data was specified, the routine continues to step **2520** to determine if category type data was specified to be included in the report. If so, the routine continues to step **2525** to retrieve stored information on occurrences of the category types that occurred during the date range. After step **2525**, or if no category type data was requested, the routine continues to step **2530** to retrieve any other types of indicated data for the requested report (e.g., administrative information or information stored about the use of exclusion definitions). The routine then continues to step **2535** to generate the requested report using the retrieved information, and provides the report to the requester (e.g., by sending a web page containing the report to the requester). The routine then continues to step **2540** to determine if more reports are to be generated. If so, the routine returns to step **2505**, and if not, the routine continues to step **2595** and ends.

In some embodiments, the routine is provided by a web server for a company acting as an Application Service Provider for one or more customers, in which the services provided include processing of interaction data for the customer and/or providing reports using process interaction data. In particular, remote customers (e.g., over the Internet) can access the web server in some embodiments and obtain reports related to their own interaction data that have previously been provided to the ASP company for processing. While not illustrated in this embodiment, in other embodiments security measures can be employed to ensure that a requester is authorized to receive the requested data and that the requested data is not inadvertently made available to others.

FIG. **26** is a flow diagram illustrating an embodiment of the Generate Data Parsing Information For Customer Content Set routine **2600**. The routine receives a content set for which interaction data will be processed (e.g., a web site whose navigation data is to be processed) or other information related to the content set, and analyzes the content set in order to generate data parsing information related to the content set. The routine begins at step **2605** where an indication of the customer content set is received. The routine continues to step **2610** where the content set is processed in such a manner as to track the relationships between different members of the content set. For example, if the content set is a web site, processing begins at the home web page for the web site, and the various links on the web pages of the web site are variously followed (or "crawled") to identify all of the available web pages and the relationships indicating what web pages have links to what other web pages.

The routine then continues to step **2615** to identify content set items that correspond to event types of interest if possible. It may be possible to classify the content set as being a member of one or more types of known content sets that have event types known to be of interest. For example, if the content set is a merchant web site that includes shopping cart web pages or other mechanisms for ordering and purchasing items, events can be defined for any such ordering-related web pages of the content set. Alternately, event types can be defined in other ways, such as defining an event type for every content set item (and optionally allowing a user to interactively remove event types that are not of interest), having meta-event type definitions that can be matched against the content set items in an attempt to determine if a content set item corresponds to a particular event type, defining events for sequences of content set items that are related in a specified manner (e.g., in a funnel-type relationship such that a first item must be accessed before a second item can be accessed), etc.

In step **2620**, the unique indicators for the content set item (e.g., URLs for web pages) are analyzed in order to identify groups of items that appear to be related (e.g., by sharing a common hierarchical data structure or by sharing similar query string names and values). The routine then continues to step **2625** to determine the server information for the one or more servers that provide the content set items, such as the domain names and IP addresses for web servers providing web site web pages. In step **2630**, the routine then generates data parsing information reflecting identified servers and their corresponding indicators, content set items corresponding to events of interest, hierarchical relationships of content set items, and/or grouping information for related items. The routine next continues to step **2635** to store the generated data parsing information in a manner that is associated with the customer and the content set. In step **2640**, it is determined whether there are more content sets for which to generate data parsing information, and if so the routine returns to step **2605**. If not, the routine continues to **2695** and ends.

While in the illustrated embodiment the routine generates data parsing information in a fully automated manner, those skilled in the art will appreciate that in other embodiments the routine could be executed in a semi-automated manner as part of a user interface by which a user is generating data parsing information for a content set. For example, the routine could perform automated processing to generate suggestions or possibilities for different types of data parsing information, and then allow the user to select or edit the generated data parsing information. Alternately, the user could perform initial preprocessing to assist the routine in generating the data parsing information, such as identifying one or more types of information about the content set (e.g., a merchant web site to assist in identifying merchant-related events of interest, or that the content set items are stored in a hierarchical manner that should be used to generate category information). In addition, the routine could generate the data parsing information in various formats, such as XML, SQL statements, etc. Moreover, the routine could generate the data parsing information to be stored and used by the parser component in a machine-readable form, but could present the same information to the user in a more human-friendly format. In some situations, such a UI could be used by a customer to themselves define and/or maintain the data parsing information for their own web site, while in other embodiments the UI is used by a trained operator of a company acting as an ASP for customers.

In addition, in some embodiments the routine can automatically generate version data for the generated data parsing information, such as by initially specifying that all of the generated data parsing information has an effective date range beginning as of the date of generation (or some other user-specified date) and having no specified end date. If the routine is later used to modify already existing data parsing information (whether user-generated or previously generated by the routine), such as in response to changes in the content set, the user could use the modification date as the beginning date for any newly generated data parsing information and use the date as the ending effective date for any data parsing information that no longer applies to the revised content set.

Those skilled in the art will also appreciate that in some embodiments the functionality provided by the routines

discussed above may be provided in alternate ways, such as being split among more routines or consolidated into less routines. Similarly, in some embodiments illustrated routines may provide more or less functionality than is described, such as when other illustrated routines instead lack or include such functionality respectively, or when the amount of functionality that is provided is altered. Those skilled in the art will also appreciate that the data structures discussed above may be structured in different manners, such as by having a single data structure split into multiple data structures or by having multiple data structures consolidated into a single data structure. Similarly, in some embodiments illustrated data structures may store more or less information than is described, such as when other illustrated data structures instead lack or include such information respectively, or when the amount or types of information that is stored is altered.

From the above description it will be appreciated that although specific embodiments of the technology have been described for purposes of illustration, various modifications may be made without deviating from the spirit and scope of the invention. For example, the processing of the parser may be performed by the data collection component before sending the data to the data warehouse server. Accordingly, the invention is not limited except by the appended claims. In addition, while certain aspects of the invention are presented below in certain claim forms, the inventors contemplate the various aspects of the invention in any available claim form. For example, while only some aspects of the invention may currently be recited as being embodied in a computer-readable medium, other aspects may likewise be so embodied. Accordingly, the inventors reserve the right to add additional claims after filing the application to pursue such additional claim forms for other aspects of the invention.

What is claimed is:

1. A computer-implemented method for using predefined parsing information to analyze web site navigation data in order to identify occurrences corresponding to defined category types, the method comprising:

for each of multiple distinct web sites each having multiple web pages,

receiving web site navigation data associated with the web site that has multiple entries each containing information about a request for a web page of the web site and a response to the request, the request including an indicated URL and sent to a web site server for the web site, each of the indicated URLs optionally including a URL path portion and optionally including a query string portion, the contained information about each request including any URL path portion that is included in the URL indicated for the request and including any query string portion that is included in the URL indicated for the request;

retrieving predefined parsing information associated with the web site that includes multiple distinct definitions of logical sites and multiple distinct definitions of category types, each logical site definition specifying an IP address and port number used by a web site server to provide at least some of the web pages of the web site, and each category type definition specifying one of the logical sites and indicating multiple web pages of the web site of that category type by including a URL pattern that is capable of matching the URL paths of the URLs corresponding to the multiple web pages and a query string pattern capable of matching the query strings of those corresponding URLs;

for each entry of the received web site navigation data,

analyzing the information contained in the entry to determine if the web site server to which the request was sent matches any of the defined logical sites by using the IP address and the port number specified by that logical site; and

if a logical site is determined to match the web site server, further analyzing the information contained in the entry by storing, for at least one of the category types whose definition specifies the matching logical site, an indication of an occurrence of a request for a web page of that one category type if the information contained in the entry includes a URL path portion that matches the URL pattern included in that one category type definition and includes a query string portion that matches the query string pattern included in that one category type definition;

receiving a request from an operator of the web site to provide information for the web site about occurrences of requests for web pages of specified category types;

retrieving in response the stored indications of the occurrences of the requests for web pages of the specified category types; and

providing the retrieved information to the operator, so that the operators of the multiple web sites can receive information about occurrences of requests for web pages of category types of interest for their web sites.

2. The method of claim 1 wherein the web pages of one of the web sites are inter-linked in a hierarchical manner having multiple hierarchy members at multiple hierarchy levels, and wherein the category types defined for that web site correspond to the multiple hierarchy members.

3. The method of claim 1 wherein the URL path portions of the URLs that correspond to the web pages of one of the web sites indicate multiple hierarchy members, and wherein the category types defined for that web site correspond to the multiple hierarchy members.

4. The method of claim 1 wherein the query string portions of the URLs that correspond to the web pages of one of the web sites include multiple values for a query string parameter name, and wherein the category types defined for that web site correspond to the multiple values.

5. The method of claim 1 wherein each of the category type definitions includes a unique name for that category type, and wherein the unique name for at least one of category types is dynamically generated using information contained in at least one entry corresponding to a request for a web page of that one category type.

6. The method of claim 1 wherein the category types for at least one of the web sites are hierarchically structured such that, for each of the category types defined for that one web site, the web pages of the web site that correspond to that category type include the web pages of the web site that correspond to each of the category types at a next lower-level in the hierarchy structure.

7. The method of claim 1 wherein at least some of the category type definitions indicate multiple web pages by including multiple page type definitions that each specify a distinct combination of a URL pattern and a query string pattern, and wherein a request for a web page that is indicated with a URL is determined to be an occurrence of a request for a web page of a category type if any URL path for the indicated URL and any query string for the indicated URL match the URL pattern and the query string pattern of any of the multiple page type definitions of that category type.

8. The method of claim 1 wherein the receiving of the web site navigation data associated with a web site includes retrieving at least one log file from at least one web site server for the web site, the retrieved log files containing the web site navigation data.

9. The method of claim 1 including, before the receiving of the web site navigation data for a web site, generating the parsing information associated with the web site based at least in part on the web site servers that can provide web pages of the web site and on the types of web pages that those web site servers can receive.

10. The method of claim 1 wherein the received request from an operator of one of the web sites further specifies effective dates such that the information to be provided is for occurrences of the specified category types that took place during the effective dates, and wherein the stored indications that are retrieved in response are for those occurrences.

11. The method of claim 1 wherein the operator of one of the web sites from whom a request is received is at a remote location, and wherein the providing of the retrieved information to the operator includes generating a report that includes the retrieved information and sending the generated report to the remote location for presentation to the operator.

12. The method of claim 1 wherein the operators of the multiple web sites are customers, and wherein the analyzing of the web site navigation data entries for the web sites is performed as a service for the customers.

13. A computer-implemented method for analyzing interaction data to identify occurrences corresponding to defined groups of related items, the method comprising:

receiving an indication of interaction data that is associated with a content set having multiple items, the interaction data having one or more entries that are each related to an interaction with at least one of the items of the content set;

receiving an indication of at least one communication definition that specifies a manner of communicating content set item interactions;

receiving an indication of multiple category type definitions each specifying a group of related content set items and each associated with one of the communication definitions, and

for each entry of the interaction data,

determining whether the entry matches one of the category type definitions in such a manner that the related interaction for the entry is with a content set item that is a member of the group specified by that category type definition and was communicated in the manner specified by the communication definition associated with that category type definition; and

when it is determined that the entry matches one of the category type definitions, storing an indication of an occurrence of that category type.

14. The method of claim 13 wherein the content set is a web site with multiple web pages, and wherein the items of the content set are the web pages.

15. The method of claim 13 wherein the content set is a group of multiple related web pages that are a subset of web pages of a web site, and wherein the items of the content set are the multiple related web pages.

16. The method of claim 13 wherein the content set is multiple related web sites each having multiple web pages, and wherein the items of the content set are the web pages of the multiple related web sites.

17. The method of claim 13 wherein the content set is a service providing multiple features, and wherein the items of the content set are the multiple features.

18. The method of claim 13 wherein the content set is an executing program providing various functionalities, and wherein the items of the content set are the various functionalities.

19. The method of claim 13 wherein the group of related content items specified for each of the category type definitions is a subset of the items for the content set.

20. The method of claim 13 wherein each of the interactions related to the interaction data entries includes specifying a Uniform Resource Indicator.

21. The method of claim 13 wherein each of the interactions related to the interaction data entries includes requesting that functionality be provided.

22. The method of claim 13 wherein each of the interactions related to the interaction data entries includes supplying information.

23. The method of claim 13 wherein the manner of communicating content set interactions specified by each of the communication definitions includes using a specified IP address and port number to communicate information related to an interaction.

24. The method of claim 13 wherein the manner of communicating content set interactions specified by each of the communication definitions includes using a specified domain name to communicate information related to an interaction.

25. The method of claim 13 wherein the manner of communicating content set interactions specified by each of the communication definitions includes using a specified group of communication parameters to communicate information related to an interaction.

26. The method of claim 13 wherein the manner of communicating content set interactions specified by each of the communication definitions includes identifying a specified portion of the content set to which an interaction is to be communicated.

27. The method of claim 13 wherein the manner of communicating content set interactions specified by each of the communication definitions includes identifying a specified computing device or computer program provider to which an interaction is to be communicated.

28. The method of claim 13 wherein the content set items are stored in multiple directories, and wherein the groups of related content set items specified for the category type definitions are the content set items stored in each of the multiple directories.

29. The method of claim 13 wherein the content set is a web site with a home web page having links each corresponding to groups of web pages of the web site, and wherein the groups of related content set items specified for the category type definitions are the groups of web pages.

30. The method of claim 13 wherein each of the content set items has an associated URL with a path portion that can include one or more hierarchical members, and wherein each group of related content set items specified for a category type definition includes content set items having a common hierarchical member in the path portion of the URL associated with the content set item.

31. The method of claim 13 wherein each of the content set items has an associated URL with a query string portion that includes a common query parameter name and corresponding value, and wherein each group of related content set items specified for a category type definition includes content set items having a common corresponding value for the common query parameter name.

32. The method of claim 13 wherein the content set items are each associated with a type of product, and wherein the

content set items in each group are related based on the product types associated with those content set items.

33. The method of claim 13 wherein the content set items are each associated with one or more features, and wherein the content set items in each group are related based on the features associated with those content set items.

34. The method of claim 13 wherein the items of the content set are organized into a hierarchical structure having multiple hierarchy levels and at least one hierarchy member at each hierarchy level, each of the content set items associated with one of the hierarchy members, and wherein each category type corresponds to one of the multiple hierarchy members such that the group of related content set items for that category type includes the content set items associated with that one hierarchy member.

35. The method of claim 34 wherein each group of related content set items for a category type corresponding to a hierarchy member further includes the content set items associated with each of the hierarchy members below that hierarchy member in the hierarchical structure.

36. The method of claim 34 wherein the hierarchical structure of the content set items is based on a hierarchical manner in which the content set items are stored.

37. The method of claim 34 wherein each of the content set items has an associated URL with a path portion, and wherein the hierarchical structure of the content set items is based on a hierarchical structure of the path portions of the associated URLs.

38. The method of claim 34 wherein each of the content set items has an associated URL with a query string portion that includes at least one query parameter name and corresponding value, and wherein the hierarchical structure of the content set items is based on the values in the query string portions.

39. The method of claim 34 wherein the content set is a web site with a home web page and with multiple other web pages accessible either directly from the home web page or indirectly from the home web page via one or more intervening other web pages, the hierarchical structure such that each of the web pages is a hierarchy member and each of the hierarchy levels includes web pages accessible from the home web page via a same number of other intervening web pages.

40. The method of claim 13 wherein each of the interactions related to the interaction data entries includes specifying a Uniform Resource Indicator, wherein each of the content set items has an associated Uniform Resource Indicator, and wherein each of the category type definitions includes a pattern capable of matching at least one Uniform Resource Indicator, the group of related content set items for a category type definition being the content set items whose associated Uniform Resource Indicator matches the included pattern for that category type definition.

41. The method of claim 13 wherein each of the interactions related to the interaction data entries includes specifying a URL corresponding to a content set item, the specified URL optionally having a path portion and optionally having a query string portion, each of the query string portions including one or more combinations each having a query parameter name and corresponding query value, and wherein each of the category type definitions includes a URL path pattern capable of matching one or more URL paths and includes a query string pattern capable of matching at least one query string, the group of related content set items for a category type definition being the content set items having corresponding URLs that match the included URL path pattern and query string pattern for that category type definition.

42. The method of claim 41 wherein the query string patterns each indicate one or more query parameter names whose presence in a query string is required, allowed, or disallowed if that query string is to match the query string pattern, and wherein the determination of whether a query string portion matches a query string pattern further includes determining if the query string portion includes each of the query parameter names whose presence is indicated in the query string pattern to be required and does not include any of the query parameter names whose presence is indicated in the query string pattern to be disallowed.

43. The method of claim 41 wherein at least some of the category type definitions include multiple page type patterns that each include a URL path pattern and a query string pattern, and wherein the related interaction for an entry is determined to be with a content set item that is a member of the group specified by a category type definition if the path portion and the query string portion of the URL corresponding to that content set item match the URL path pattern and the query string pattern of any of the page type patterns for that category type definition.

44. The method of claim 41 wherein at least some of the URL path patterns include a static portion capable of matching a single corresponding portion of a URL path and include a variable portion capable of matching multiple corresponding portions of URL paths.

45. The method of claim 41 wherein each of the URL path patterns can be specified to match any URL path and wherein each of the query string patterns can be specified to match any query string.

46. The method of claim 13 wherein each of the category type definitions includes a unique name for that category type, and wherein the unique name for at least one of category types is dynamically generated using information from at least one interaction with a content set item that is a member of the group specified by that category type definition.

47. The method of claim 46 wherein each of the interactions related to the interaction data entries includes specifying a URL corresponding to a content set item, the specified URL having a path portion and a query string portion, the query string portion including at least one query parameter name and corresponding query value, and wherein the information used to dynamically generate the unique name is at least one query value for the specified URL corresponding to the content set item that is a member of the group specified by that category type definition.

48. The method of claim 13 wherein at least some of the groups of related content set items contain a single content set item.

49. The method of claim 13 wherein each of the communication definitions can be specified to match any manner of communicating content set interactions.

50. The method of claim 13 wherein each of the category type definitions can be specified to include any content set item.

51. The method of claim 13 wherein at least some of the entries match multiple of the category type definitions.

52. The method of claim 51 wherein an indication of an occurrence is stored for each of the multiple category type definitions.

53. The method of claim 51 wherein a single indication of an occurrence is stored for the one of the multiple category type definitions having a highest degree of match to the entry.

54. The method of claim 51 wherein the category type definitions are hierarchically structured such that the group

of related content set items for a category type definition includes the content set items in the groups of related content set items for each of the category type definitions below that category type definition in the hierarchical structure, and wherein a single indication of an occurrence is stored for the one of the multiple category type definitions that is lowest in the hierarchical structure.

55. The method of claim **51** wherein the category type definitions are hierarchically structured such that the group of related content set items for a category type definition includes the content set items in the groups of related content set items for each of the category type definitions below that category type definition in the hierarchical structure, and wherein a single indication of an occurrence is stored for the one of the multiple category type definitions that is highest in the hierarchical structure.

56. The method of claim **13** wherein each of the entries contain information related to the interaction for the entry, and wherein the determining that an entry matches a category type definition includes analyzing the information contained in the entry.

57. The method of claim **13** including receiving a request to provide information about occurrences of specified category types, and providing in response the stored indications of occurrences related to the specified category types.

58. The method of claim **13** wherein the determining of whether the interaction data entries match category type definitions is performed as a service for a customer.

59. A computer-readable medium whose contents cause a computing device to analyze data to identify occurrences corresponding to defined groups of items, by performing a method comprising:

receiving an indication of data that is associated with a content set having multiple items, the data having one or more entries that are each related to an interaction with at least one of the items of the content set;

receiving an indication of multiple definitions each specifying a group of related content set items; and

for each entry of the data,

determining whether the entry matches one of the definitions in such a manner that the related interaction for the entry is with a content set item that is a member of the group specified by that definition; and

when it is determined that the entry matches one of the definitions, indicating an occurrence of an interaction with the group of items specified by that definition.

60. The computer-readable medium of claim **59** wherein the computer-readable medium is a memory of a computer system.

61. The computer-readable medium of claim **59** wherein the computer-readable medium is a data transmission medium transmitting a generated data signal containing the contents.

62. The computer-readable medium of claim **59** wherein the contents are instructions that when executed cause the computing device to perform the method.

63. A computing device for analyzing interaction data to identify occurrences corresponding to defined category types, comprising:

an interaction data receiver component capable of receiving an indication of interaction data that is associated with a content set having multiple items, the interaction data having one or more entries that are each related to an interaction with at least one of the items of the content set;

a definition receiver component capable of receiving an indication of at least one communication definition that specifies a manner of communicating content set item interactions and of receiving an indication of multiple category type definitions each specifying a group of related content set items and each associated with one of the communication definitions; and

an interaction data parsing component capable of, for each entry of the interaction data, determining whether the entry matches one of the category type definitions in such a manner that the related interaction for the entry is with a content set item that is a member of the group specified by that category type definition and was communicated in the manner specified by the communication definition associated with that category type definition and of storing an indication of an occurrence of a category type when it is determined that an entry matches the definition for that category type.

64. The computing device of claim **63** wherein the interaction data receiver component, definition receiver component and interaction data parsing component are executing in memory of the computing device.

65. A computer system for analyzing interaction data to identify occurrences corresponding to defined category types, comprising:

means for receiving an indication of interaction data that is associated with a content set having multiple items, the interaction data having one or more entries that are each related to an interaction with at least one of the items of the content set;

means for receiving an indication of at least one communication definition that specifies a manner of communicating content set item interactions and for receiving an indication of multiple category type definitions each specifying a group of related content set items and each associated with one of the communication definitions; and

means for, for each entry of the interaction data, determining whether the entry matches one of the category type definitions in such a manner that the related interaction for the entry is with a content set item that is a member of the group specified by that category type definition and was communicated in the manner specified by the communication definition associated with that category type definition, and for storing an indication of an occurrence of a category type when it is determined that an entry matches the definition for that category type.

66. A computer-implemented method for analyzing interaction data for a web site to identify occurrences corresponding to defined category types, the method comprising:

receiving an indication of multiple interaction data entries each containing information about an interaction with a web site that includes a specified URL corresponding to one of multiple web pages of the web site, each of the specified URLs optionally including a URL path portion and optionally including a query string portion, the contained information for each entry including any URL path portion that is included in the specified URL for the entry and including any query string portion that is included in the specified URL for the entry;

receiving an indication of multiple category type definitions that each specify a group of web pages related to a category by using a URL path pattern capable of matching at least one URL path related to the category and using a query string pattern capable of matching at least one query string related to the category; and

for each entry,

analyzing the entry to determine whether the entry matches one of the category type definitions by containing information about a specified URL corresponding to a web page that is related to the category for that category type definition, the matching such that the contained information includes a URL path portion that matches the URL path pattern for that one category type definition and includes a query string portion that matches the query string pattern for that one category type definition; and

when it is determined that the entry matches one of the category type definitions, storing an indication of an occurrence of that category type for the web site.

67. The method of claim **66** wherein the contained information about each interaction further includes information related to a manner of identifying a web site server with which the interaction occurred, wherein each of the category type definitions is associated with a logical site definition that specifies a manner of identifying a web site server related to the web site, and wherein the determining that an entry matches a category type definition further includes determining that the information contained in the entry that is related to the manner of identifying the web site server matches the manner of identifying a web site server specified by the logical site definition associated with that category type definition.

68. The method of claim **67** wherein the manner of identifying a web site server related to the web site that is specified by each logical site definition includes using a specified IP address and port number to communicate with the web site server.

69. The method of claim **66** wherein each of the interactions with a web site that includes a specified URL includes a request for a web page from that web site that corresponds to the specified URL.

70. The method of claim **66** wherein each of the interactions with a web site that includes a specified URL includes a sending to a client of a web page from that web site that corresponds to the specified URL.

71. The method of claim **66** wherein each of the web pages has an associated URL with a path portion that can include one or more hierarchical members, and wherein each group of web pages specified for a category type definition includes web pages having a common hierarchical member in the path portion of the URL associated with the web page.

72. The method of claim **66** wherein each of the web pages has an associated URL with a query string portion that includes a common query parameter name and corresponding value, and wherein each group of related web pages specified for a category type definition includes web pages having a common corresponding value for the common query parameter name.

73. The method of claim **66** wherein the web site is organized into a hierarchical structure having multiple hierarchy levels and at least one hierarchy member at each hierarchy level, each of the web pages associated with one of the hierarchy members, and wherein each category type corresponds to one of the multiple hierarchy members such that the group of web pages for that category type includes the web pages associated with that one hierarchy member.

74. The method of claim **73** wherein each group of web pages for a category type corresponding to a hierarchy member further includes the web pages associated with each of the hierarchy members below that hierarchy member in the hierarchical structure.

75. The method of claim **73** wherein the web site has a home web page such that the other web pages are accessible either directly from the home web page or indirectly from the home web page via one or more intervening other web pages, the hierarchical structure such that each of the web pages is a hierarchy member and each of the hierarchy levels includes web pages accessible from the home web page via a same number of other intervening web pages.

76. The method of claim **66** wherein each of the URL path patterns can be specified to match any URL path and wherein each of the query string patterns can be specified to match any query string.

77. The method of claim **66** wherein at least some of the groups of web pages contain a single web page.

78. The method of claim **66** wherein at least some of the entries match multiple of the category type definitions.

79. The method of claim **78** wherein an indication of an occurrence is stored for each of the multiple category type definitions.

80. The method of claim **66** including receiving a request to provide information about occurrences of specified category types, and providing in response the stored indications of occurrences related to the specified category types.

81. A computer-readable medium containing instructions that when executed cause a computer system to analyze data to identify occurrences corresponding to defined groups of web pages, by performing a method comprising:

receiving an indication of multiple data entries each containing information about an interaction with a web site that includes a specified URL corresponding to one of multiple web pages of the web site, each of the specified URLs optionally including a URL path portion and optionally including a query string portion, the contained information for each entry including any URL path portion that is included in the specified URL for the entry and including any query string portion that is included in the specified URL for the entry;

receiving an indication of multiple definitions that each specify a group of web pages related to a category by using a URL path pattern capable of matching at least one URL path related to the category and using a query string pattern capable of matching at least one query string related to the category; and

for each entry,

analyzing the entry to determine whether the entry matches one of the definitions by containing information about a specified URL corresponding to a web page that is related to the category for that definition, the matching such that the contained information includes a URL path portion that matches the URL path pattern for that one definition and includes a query string portion that matches the query string pattern for that one definition; and

when it is determined that the entry matches one of the definitions, indicating an occurrence of an interaction with the group of web pages specified by that definition.

82. A computer-implemented method for analyzing interaction data for a web site to identify occurrences corresponding to defined category types, the method comprising:

receiving an indication of multiple interaction data entries each containing information about a request that specifies a URL corresponding to a web page of a web site, each of the specified URLs optionally including a URL path portion and optionally including a query string portion, each of the query string portions including one or more combinations each having a query parameter name and corresponding query value, the contained information about each request including any URL path

portion that is included in the specified URL for the request and including any query string portion that is included in the specified URL for the request;

receiving an indication of a category type definition corresponding to multiple categories, the category type definition specifying a URL path pattern capable of matching at least one URL path related to the multiple categories and a query string pattern capable of matching at least one query string related to the multiple categories, each query string pattern indicating one or more query parameter names, the category type definition further specifying a name definition for providing a unique name for each of the multiple categories, the name definition including at least one of the indicated query parameter names and indicating how values for each of the included query parameter names are to be combined to form the names of the multiple categories, each unique combination of values for the indicated query parameter names corresponding to one of the multiple categories; and

for each entry,

analyzing the entry to determine whether the entry matches one of the category type definitions by containing information about a request corresponding to a web page that is related to the category for that category type definition, the matching such that the information contained in the entry includes a URL path portion that matches the URL path pattern specified in that one category type definition and includes a query string portion that matches the query string pattern specified in that one category type definition; and

when it is determined that the entry matches one of the category type definitions,

determining the name of the category to which the entry corresponds by retrieving the value from the query string portion of the contained information for the entry that corresponds to each of the query parameter names included in the name definition and by combining the retrieved values in the manner indicated in the name definition; and

storing an indication of an occurrence of the category having the name formed by the combined retrieved values.

83. The method of claim **82** wherein the contained information about each request further includes information related to a manner of identifying a web site server to which the request was sent, wherein each of the category type definitions is associated with a logical site definition that specifies a manner of identifying a web site server related to the web site and wherein the determining that an entry matches a category type definition further includes determining that the information included in the entry that is related to the manner of identifying the web site server matches the manner of identifying a web site server specified by the logical site definition associated with that category type definition.

84. A computer-implemented method for analyzing interaction data for a web site to identify occurrences corresponding to defined category types, the method comprising:

receiving an indication of multiple interaction data entries each containing information about a request that specifies a URL corresponding to a web page of a web site, each of the specified URLs optionally including a URL path portion and optionally including a query string portion, the contained information about each request including any URL path portion that is included in the specified URL for the request and including any query string portion that is included in the specified URL for the request;

receiving an indication of multiple category type definitions that each specify a group of web pages related to a category with multiple page type patterns that each specify a distinct combination of a URL path pattern capable of matching at least one URL path related to the category and a query string pattern capable of matching at least one query string related to the category; and

for each entry,

analyzing the entry to determine whether the entry matches one of the category type definitions by containing information about a request corresponding to a web page that is related to the category for that category type definition, the matching such that, for any of the page type patterns included in that one category type, definition, the information contained in the entry includes a URL path portion and a query string portion that match the URL path pattern and the query string pattern specified in that page type pattern; and

when it is determined that the entry matches one of the category type definitions, storing an indication of an occurrence of that category type for the web site.

85. The method of claim **84** wherein the contained information about each request further includes information related to a manner of identifying a web site server to which the request was sent, wherein each of the category type definitions is associated with a logical site definition that specifies a manner of identifying a web site server related to the web site, and wherein the determining that an entry matches a category type definition further includes determining that the information included in the entry that is related to the manner of identifying the web site server matches the manner of identifying a web site server specified by the logical site definition associated with that category type definition.

86. A computer-implemented method for analyzing interaction data for a web site to identify occurrences corresponding to defined category types, the method comprising:

receiving an indication of multiple interaction data, entries each containing information about a request that specifies a URL corresponding to a web page of the web site, each of the specified URLs optionally including a URL path portion and optionally including a query string portion, each of the query string portions including one or more combinations each having a query parameter name and corresponding query value, the contained information about each request including any URL path portion that is included in the specified URL for the request and including any query string portion that is included in the specified URL for the request;

receiving an indication of multiple category type definitions that each specify a group of web pages related to a category with a URL path pattern capable of matching at least one URL path related to the category and a query string pattern capable of matching at least one query string related to the category, each query string pattern indicating one or more query parameter names whose presence in a query string is required, allowed, or disallowed if that query string is to match the query string pattern; and

for each entry,

analyzing the entry to determine whether the entry matches one of the category type definitions by containing information about a request corresponding to a web page that is related to the category for that category type definition, the matching such that the information contained in the entry

(a) includes a URL path portion that matches the URL path pattern specified in that one category type definition and

(b) includes a query string portion that includes each of the query parameter names whose presence is indicated in the query string pattern specified in that one category type definition to be required, and that does not include any of the query parameter names whose presence is indicated in the query string pattern specified in that one category type definition to be disallowed; and

when it is determined that the entry matches one of the category type definitions, storing an indication of an occurrence of that one category type for the web site.

87. The method of claim **86** wherein the contained information about each request further includes information related to a manner of identifying a web site server to which the request was sent, wherein each of the category type definitions is associated with a logical site definition that specifies a manner of identifying a web site server related to the web site, and wherein the determining that an entry matches a category type definition further includes determining that the information included in the entry that is related to the manner of identifying the web site server matches the manner of identifying a web site server specified by the logical site definition associated with that category type definition.

88. The method of claim **86** wherein each of the query string patterns additionally indicates a type of value corresponding to at least some of the indicated query parameter names, and wherein the determining that an entry matches a category type definition further includes determining that, for each of the query parameter names that is included in the query string portion of the contained information for the entry and that is indicated to have a type of value in the query string pattern specified by that one category type definition, the corresponding query value in the query string portion matches the indicated value type.

89. A computer-implemented method for analyzing interaction data to identify occurrences corresponding to defined hierarchies of items, the method comprising:

receiving an indication of multiple interaction data entries each containing information related to an interaction with one of multiple items of a content set, the content set items structured in a hierarchy having multiple hierarchy members at multiple hierarchy levels;

receiving an indication of multiple hierarchy definitions that each correspond to one or more related hierarchy members; and

for each entry,

analyzing the entry to determine whether the entry matches one of the hierarchy definitions by containing information about an interaction with a content set item that is one of the hierarchy members to which that one hierarchy definition corresponds; and

when it is determined that the entry matches one of the hierarchy definitions, indicating an occurrence of an interaction with the related hierarchy members to which that one hierarchy definition corresponds.

90. The method of claim **89** wherein the content set items are web pages of a web site, wherein each of the interaction data entries contains a URL specified as part of a request corresponding to one of the web pages, each specified URL optionally including a URL path portion and optionally including a query string portion, wherein each of the hierarchy member definitions includes a URL path pattern capable of matching at least one URL path related to the one or more web pages corresponding to the hierarchy member and includes a query string pattern capable of matching at least one query string related to the one or more web pages corresponding to the hierarchy member, and wherein the matching of a hierarchy definition to an entry is based on the URL contained in the entry including a URL path portion that matches the URL path pattern for that one hierarchy member definition and including a query string portion that matches the query string pattern for that one hierarchy member definition.

91. The method of claim **90** wherein the contained information about each request further includes information related to a manner of identifying a web site server to which the request was sent, wherein each of the hierarchy member definitions is associated with a logical site definition that specifies a manner of identifying a web site server related to the web site, and wherein the determining that an entry matches a hierarchy member definition further includes determining that the information included in the entry that is related to the manner of identifying the web site server matches the manner of identifying a web site server specified by the logical site definition associated with that category type definition.

92. The method of claim **89** wherein the content set items are structured in the hierarchy based on product types associated with the content set items.

93. The method of claim **89** wherein the content set items are structured in the hierarchy based on features associated with the content set items.

94. The method of claim **89** wherein the content set items are structured in the hierarchy based on categories associated with the content set items.

95. A computer-implemented method for analyzing usage data to identify occurrences corresponding to defined groups of features, the method comprising:

receiving an indication of usage data associated with a provided service or an executing computer program that has multiple features available for use, the usage data having multiple entries each related to a distinct use of one of multiple features of the provided service or executing computer program that includes information being communicated;

receiving an indication of multiple definitions each specifying a group of features related to a category and each associated with a manner of communicating information to the provided service or to the executing computer program; and

for each entry of the usage data,

determining whether the entry matches one of the definitions such that the related use for the entry is of a feature that is a member of the group of features specified by that definition and such that the information communicated for the related use is communicated in the manner associated with that definition, and

when it is determined that the entry matches one of the definitions, storing an indication of an occurrence of a use of the group of features specified by that definition.

96. A computer-readable medium containing a data structure that stores multiple definitions for category types so that

occurrences of those category types in interaction data for a web site can be identified, the data structure having multiple entries each corresponding to a category type definition that specifies a group of web pages related to a category, each entry storing a URL path pattern capable of matching at least one URL path related to the category and a query string pattern capable of matching at least one query string related to the category,

such that when analyzing information about an interaction with a web page of the web site having a specified URL that optionally includes a URL path portion and optionally includes a query string portion, if the web page is determined to be a member of the group of web pages specified by a category type definition then an occurrence of that category type is indicated, the web page determined to be a member of the group of web pages for a category type definition if the specified URL includes a URL path portion that matches the URL path pattern specified for that category type definition and includes a query string portion that matches the query string pattern specified for that category type definition.

97. The computer-readable medium of claim **96** wherein each of the entries further includes an indication of a logical site definition that specifies a manner of identifying a web site server related to the web site,

such that, when the information about the interaction further includes information related to a manner of identifying a web site server with which the interaction occurred, the web page is determined to be a member of the group of web pages for a category type definition only if the information related to the manner of identifying the web site server matches the manner of identifying a web site server specified by the logical site definition indicated by that category type definition.

98. The computer-readable medium of claim **96** wherein the category type definitions are related to each other and wherein at least some of the entries further include an indication of a relationship of the category type definition for that entry to at least one other category type definition,

such that, when the web page is determined to be a member of the group of web pages for a category type definition whose entry includes an indication of a relationship to other category type definitions, the web page is also determined to be a member of the group of web pages for at least some of the other related category type definitions.

99. The computer-readable medium of claim **96** wherein the category type definitions corresponding to at least some of the entries each have multiple distinct combinations of a URL path pattern and a query string pattern, the entry for each of those category type definitions further storing the multiple combinations of URL path patterns and query string patterns of the patterns for that category type definition,

such that the web page is determined to be a member of the group of web pages for a category type definition having multiple combinations if, for any of those combinations, the information includes a URL path portion that matches the URL path pattern specified in that combination and includes a query string portion that matches the query string pattern specified in that combination.

100. The computer-readable medium of claim **96** wherein the stored query string patterns each indicate one or more query parameter names whose presence in a query string is required, allowed, or disallowed if that query string is to match the query string pattern,

such that a query string portion of the information is determined to match the query string pattern specified for one of the category type definitions if the query string portion includes each of the query parameter names whose presence is indicated in that query string pattern to be required and does not include any of the query parameter names whose presence is indicated in that query string pattern to be disallowed.

101. The computer-readable medium of claim **96** further containing a data structure having multiple entries that each store an exclusion definition that specifies a type of interaction,

such that if the information being analyzed is of a type matching one of the exclusion definitions, the information will not be determined to match any of the category type definitions.

102. A computer-readable medium containing a data structure storing multiple definitions for category types so that occurrences of those category types can be identified in interaction data or usage data, the data structure having multiple entries each corresponding to a category type definition, each entry specifying a group of related content set items for a content set and including an indication of a communication definition that specifies a manner of communicating information related to interactions or uses corresponding to the content set items,

so that when analyzing data about an interaction or use that corresponds to a content set item and that indicates a manner in which related information was communicated, if the data matches one of the category type definitions in such a manner that the interaction or use corresponds to one of the content set items in the group specified by that category type definition and had related information that was communicated in the manner specified by the communication definition indicated by that category type definition, an occurrence of that category type can be identified.

103. A method for analyzing customer data to identify occurrences corresponding to defined categories, the method comprising:

receiving a request from a customer to analyze interaction or usage data for that customer related to a content set having multiple content set items;

receiving an indication of definitions for the customer that each specify a group of content set items related to a category and are each associated with at least one manner of communicating information;

receiving a first set of data for the customer that includes information about at least one interaction or use;

analyzing the received set of data to determine whether the received data includes information about any interactions or uses that match one of the definitions in such a manner that the interaction or use is with a content set item in the group specified by that definition and had related information communicated in a manner associated with that definition; and

when it is determined that the received data matches one of the definitions, providing information to the customer about an occurrence for that category.

104. The method of claim **103** wherein sets of data are automatically retrieved from the customer and analyzed on a periodic basis.

105. The method of claim **103** wherein the providing of the information to the customer includes generating reports on a periodic basis and sending the generated reports to the customer.

106. The method of claim 103 including storing an indication of the occurrence for that category, and wherein the providing of the information to the customer includes receiving a request from the customer at a remote location to provide information about occurrences of one or more categories and sending the requested information to the remote location.

107. The method of claim 103 wherein the method is performed for multiple customers each having distinct interaction or usage data and having distinct definitions.

108. A method for creating definitions of category types for analyzing interaction data for a web site to identify occurrences corresponding to defined category types, the method comprising:

receiving an indication of a log file for the web site or of other information related to the web site that indicates multiple interactions with a web site server for the web site, each indicated interaction having associated information including network address information for the web site server and a URL specified as part of the interaction;

analyzing the log file or the other information to identify distinct network addresses for the web site servers for the web site, and generating a site definition for each of the identified network addresses that includes that network address; and

analyzing the log file or the other information to identify groups of related web pages, and generating a category type definition for each of the identified groups, so that information about an interaction with the web site can be analyzed to determine whether the information matches one of the category type definitions in such a manner that the interaction is with a web page that is a member of the group specified by that category type definition and was with a web site server having a network address that matches one of the site definitions.

109. The method of claim 108 wherein the specified URLs each optionally include a URL path portion and optionally include a query string portion, and wherein the category type definitions are each generated to include a URL path pattern capable of matching at least one URL path related to the group of web pages for that category type definition and a query string pattern capable of matching at least one query string related to the group of web pages for that category type definition.

110. The method of claim 108 wherein each of the category type definitions are further generated to include an indication to one of the generated site definitions, such that information about an interaction with the web site is not determined to match one of the category type definitions unless the interaction was with the web site server having the network address included in the site definition indicated by that one category type definition.

111. The method of claim 108 including analyzing information about an interaction with the web site using the generated category type definitions and site definitions.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

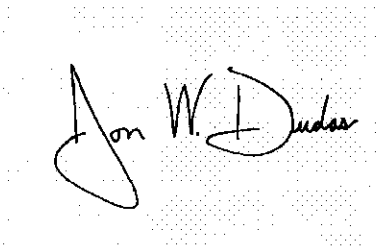
PATENT NO. : 6,917,972 B1
DATED : July 12, 2005
INVENTOR(S) : Roman Basko et al.

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Column 7,
Line 32, "report" should be -- reports --.

Signed and Sealed this

Twenty-seventh Day of September, 2005



JON W. DUDAS
Director of the United States Patent and Trademark Office